



Universidade Federal de Pernambuco
Centro de Informática

Pós-graduação em Ciência da Computação

**ANÁLISE DE DEPENDABILIDADE DE
SISTEMAS DATA CENTER BASEADA EM
ÍNDICES DE IMPORTÂNCIA**

José Jair Cavalcante de Figueirêdo

DISSERTAÇÃO DE MESTRADO

Recife
26 de agosto de 2011

Universidade Federal de Pernambuco
Centro de Informática

José Jair Cavalcante de Figueirêdo

**ANÁLISE DE DEPENDABILIDADE DE SISTEMAS DATA
CENTER BASEADA EM ÍNDICES DE IMPORTÂNCIA**

*Trabalho apresentado ao Programa de Pós-graduação em
Ciência da Computação do Centro de Informática da Uni-
versidade Federal de Pernambuco como requisito parcial
para obtenção do grau de Mestre em Ciência da Com-
putação.*

Orientador: *Prof. Dr. Paulo Romero Martins Maciel*

Recife
26 de agosto de 2011

Catálogo na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Figueirêdo, José Jair Cavalcante de
Análise de dependabilidade de sistemas data center
baseada em índices de importância / José Jair Cavalcante
de Figueirêdo - Recife: O Autor, 2011.
xiii, 88 folhas : il., fig., tab.

Orientador: Paulo Romero Martins Maciel.
Dissertação (mestrado) - Universidade Federal de
Pernambuco. CIn, Ciência da computação, 2011.

Inclui bibliografia.

1. Ciência da computação. 2. Avaliação de desempenho. 3.
Sistemas distribuídos. I. Maciel, Paulo Romero Martins
(orientador). II. Título.

004

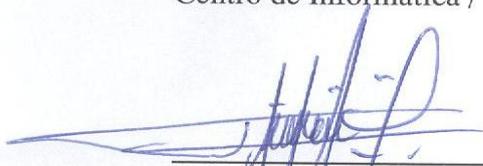
CDD (22. ed.)

MEI2011 – 125

Dissertação de Mestrado apresentada por **José Jair Cavalcante de Figueirêdo** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Análise de Dependabilidade de Sistemas Data Center Baseada em Índices de Importância**" orientada pelo **Prof. Paulo Romero Martins Maciel** e aprovada pela Banca Examinadora formada pelos professores:



Prof. Eduardo Antonio Guimarães Tavares
Centro de Informática / UFPE

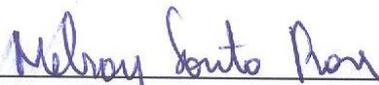


Prof. Francisco Vieira de Souza
Departamento de Informática e Estatística / UFPI



Prof. Ricardo Massa Ferreira Lima
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 26 de agosto de 2011.



Prof. Nelson Souto Rosa

Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

À minha família.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por permitir e favorecer de todas as formas a realização e conclusão deste trabalho. Por me iluminar e tornar claras as decisões que tomei no período.

Agradeço aos meus pais e aos meus irmãos pelo apoio e suporte, principalmente nos momentos de maior dificuldade, pelas horas de preocupação, pelas orações e momentos de dedicação de todos os familiares.

Agradeço a Catarina Costa, pelo incentivo, força, apoio, auxílio, por estar sempre disposta a ajudar, de alguma forma, a concretizar este trabalho, além da compreensão nos momentos que precisei me dedicar ao Mestrado. Aos amigos Gustavo Callou, Tiago Falcão, Rubens Matos, Gracieth Mendes e Bruno Silva pelas palavras de incentivo quando foi preciso, pelos momentos de descontração em meio a tanto trabalho e estudo, pelo auxílio quando precisei e, claro, pela amizade, que espero não acabe com o fim do mestrado.

Agradeço ao meu orientador, professor Paulo Maciel, que sempre esteve disponível para me orientar, confiou no meu trabalho e me incentivou.

Não poderia deixar de agradecer à Universidade Federal do Acre (UFAC), pelo incentivo à qualificação, me liberando por dois anos para poder me dedicar exclusivamente ao Mestrado. À Universidade Federal de Pernambuco, por meio do Centro de Informática, que forneceu todo o suporte necessário para a realização da pesquisa.

Finalmente, agradeço a todos aqueles, que de forma direta ou indireta contribuíram para a realização deste trabalho, que me incentivaram e apoiaram, pessoas com as quais já convivia e outras que passei a conviver durante este período.

Se não puder vencer pelo talento, vença pelo esforço.

—MAX BEERBOHM

RESUMO

Nos dias atuais, questões relacionadas à dependabilidade, como alta disponibilidade e alta confiabilidade estão cada vez mais em foco, principalmente devido aos serviços baseados na internet, que normalmente requerem operação ininterrupta dos serviços. Para melhorar os sistemas (arquiteturas de *data center*, por exemplo), deve ser realizada análise de dependabilidade. As atividades de melhoria normalmente envolvem redundância de componentes o que exige ainda, análise dos componentes. A fim de obter os valores de dependabilidade, além de entender as funcionalidades dos componentes, é importante quantificar a importância de cada componente para o sistema, além da relação entre dependabilidade e custos. Neste contexto, é importante o auxílio de ferramentas que automatizem atividades do projeto, reduzindo o tempo para se obter os resultados, quando comparados ao processo manual. Este trabalho propõe novos índices para quantificar a importância de componentes, relacionando custos, que podem auxiliar projetistas de *data center*. Adicionalmente, extensões para a ferramenta ASTRO (núcleo Mercury) foram implementadas e representam um dos resultados deste trabalho. Estas extensões incluem avaliação de importância de componentes, avaliação de dependabilidade por limites e geração das funções lógica e estrutural. Além disso, foram implementadas melhorias no módulo de Diagramas de Blocos para Confiabilidade (RBD). Mercury permite a análise de dependabilidade através de Redes de Petri Estocásticas (SPN) e RBD. Ainda considerando a ferramenta ASTRO, é possível quantificar o impacto na sustentabilidade de infraestruturas de *data center*. Todas as métricas implementadas foram avaliadas em arquiteturas de *data center*, embora não sejam limitadas a estas estruturas, podendo ser utilizadas para avaliar sistemas em geral. Para demonstrar a aplicabilidade deste trabalho foram gerados três estudos de caso.

Palavras-chave: Avaliação de Dependabilidade, Importância de Componentes, Custo, Sistemas *Data Center*, Diagrama de Bloco para a Confiabilidade.

ABSTRACT

Nowadays, concerns about dependability such as high availability and reliability are in focus, mainly due to Internet services which usually require uninterrupted operation. To improve the system (e.g., data center architectures) dependability analysis can be performed, in which component redundancy technics are usually adopted. In order to accomplish those values, besides understanding components functionality, it is important to quantify their importance to each other and to the system in relation to cost and reliability metrics. In this context, tools are important to automate many project activities, reducing the time for computing the results in comparison to manual process. This work proposes new indexes for quantifying the component importance, including cost issues, to support data center designers. Additionally, extensions for the developed environment, ASTRO tool (Mercury core), were implemented and represent one of the results of this work. Those extensions include the evaluation of component important, bounds for dependability evaluation, generation of logic and structural functions. Besides, improvements in the core Reliability Block Diagram (RBD) module were performed. Mercury allows the analysis of dependability by means of Stochastic Petri Nets (SPN) and RBD. Moreover, considering ASTRO tool, sustainability impact of data center infrastructures may be quantified. All metrics and functionalities were evaluated in data center architectures. However, the adopted methodology is general enough to analyze other systems. In addition, three case studies were conducted to demonstrate the applicability of this work.

Keywords: Dependability, Component Importance, Data Center, Cost, Reliability Block Diagram.

SUMÁRIO

Lista de Figuras	xi
Lista de Tabelas	xiii
Capítulo 1—Introdução	1
1.1 Motivação e Justificativa	3
1.2 Objetivos	4
1.3 Trabalhos Relacionados	5
1.3.1 Análise de Importância de Componentes	5
1.3.2 Ferramentas para Análise de Dependabilidade	6
1.4 Estrutura da Dissertação	8
Capítulo 2—Fundamentos	9
2.1 Infraestruturas de Data Center	9
2.2 Aspectos Básicos sobre Dependabilidade	11
2.2.1 Atributos da Dependabilidade	13
2.3 Modelos para Avaliação de Dependabilidade	14
2.3.1 Diagramas de Blocos para a Confiabilidade	15
2.4 Função de Estrutura e Função Lógica	17
2.4.1 Função de Estrutura	18
2.4.2 Função Lógica	19
2.5 Sistemas Coerentes	20
2.6 Caminhos e Cortes Mínimos	20
2.7 Métodos de Avaliação	22
2.7.1 Métodos Exatos	23

2.7.2	Métodos Aproximados	25
2.7.2.1	Método Soma dos Produtos Disjuntos	26
2.8	Importância de Componentes	28
2.8.1	Importância para Confiabilidade	30
Capítulo 3—Ferramenta ASTRO		32
3.1	Arquitetura e Funcionalidades Gerais	32
3.2	Editores de Potência e Resfriamento	33
3.2.1	Conversão de modelos de <i>data center</i> para SPN e RBD	34
3.2.2	Avaliação do Impacto na Sustentabilidade	36
3.3	Editor e Avaliador SPN	37
3.4	Editor e Avaliador RBD	39
Capítulo 4—Métodos e Métricas		41
4.1	Geração da Função Estrutural	41
4.2	Geração da Função Lógica	43
4.3	Avaliação Básica das Métricas de Dependabilidade	44
4.4	Avaliação de Dependabilidade por Limites	47
4.4.1	Geração de Caminhos e Cortes Mínimos	47
4.4.2	Obtenção dos Limites Superiores e Inferiores	49
4.4.3	Avaliação de Modelos com Componentes Repetidos	50
4.5	Redução do Modelo RBD	52
4.6	Avaliação de Importância de Componentes	54
4.6.1	Importância para a Confiabilidade	55
4.6.2	Importância para a Disponibilidade	56
4.6.3	RIc - Importância para a Confiabilidade e Custos	58
4.6.4	AIC - Importância para a Disponibilidade e Custos	61
4.7	Considerações Finais	61
Capítulo 5—Estudo de Caso		63

5.1	Metodologia para Avaliação de Importância de Componentes Considerando Custos	63
5.2	Arquitetura Base de Potência de Data Center	65
5.3	Estudo de Caso I: RI aplicada a infraestruturas de potência	66
5.4	Estudo de Caso II: RI_C aplicada em infraestruturas de potência	69
5.5	Estudo de Caso III: Avaliação de Dependabilidade por Limites	73
5.6	Considerações Finais	78
Capítulo 6—Conclusões		79
6.1	Contribuições	80
6.2	Restrições e Trabalhos Futuros	81

LISTA DE FIGURAS

2.1	Representação de um <i>Data Center</i> por subsistemas	10
2.2	Árvore da Dependabilidade. Adaptado de [Avizienis et al., 2001]	12
2.3	Relacionamentos básicos em um Diagrama de Blocos para a Confiabilidade	16
2.4	Exemplo sistema com construções série e paralelo	18
2.5	Diagrama de Blocos para Confiabilidade de sistemas coerentes com (a) caminhos mínimos e (b) cortes mínimos. Fonte [Kuo e Zuo, 2003]	21
2.6	Modelo Série/Paralelo	22
2.7	Modelo RBD com estruturas série e paralelo	24
2.8	Usando reduções série e paralelo no modelo RBD	24
2.9	Sistema com 3 componentes em série	29
2.10	Sistema com 1 componente B redundante	31
3.1	Estrutura da Ferramenta ASTRO	32
3.2	Editor de Potência na Ferramenta ASTRO	34
3.3	Importação de subsistemas na visão <i>data center</i>	35
3.4	ASTRO: Conversão para RBD e SPN	35
3.5	Variação de Parâmetro em Avaliação de Sustentabilidade	37
3.6	Resultados de Simulação na ASTRO	38
3.7	Editor de modelos RBD na ASTRO	39
3.8	Resultado de uma avaliação em RBD	40
4.1	Exemplo de Modelo RBD	42
4.2	Exemplo de Geração de Função Estrutural para o RBD da Figura 4.1	42
4.3	Exemplo de Geração de Função Lógica	43
4.4	Avaliação da função lógica com bloco falhos no modelo	44
4.5	Modelo RBD constituído por blocos em série e paralelo	45
4.6	Exemplo de Avaliação Básica em RBD	46

4.7	Experimentação com Rótulos em RBD	47
4.8	Exemplo de Avaliação por Limites/Aproximação	50
4.9	Geração de Modelo RBD com repetição de componente	51
4.10	Exemplo de Redução de Modelo RBD	54
4.11	Análise de importância para a confiabilidade	56
4.12	Resultado da análise de importância para a disponibilidade	57
4.13	RBD com 3 componentes em série	59
4.14	Modelo RBD com 4 componentes série e paralelo	60
4.15	Variação da métrica Rlc conforme variação de custo	60
5.1	Metodologia para Avaliação de Importância de Componentes	64
5.2	Arquitetura de potência base	65
5.3	Arquiteturas de Potência de Data Center Consideradas	67
5.4	Increasing with redundancy considering RI	69
5.5	Arquiteturas de Potência de Data Center Consideradas	70
5.6	Comparação das Arquiteturas Geradas com a Baseline	72
5.7	Infraestrutura com dois sistemas de potência redundantes	74
5.8	Infraestruturas de potência A2 (a) e A3 (b)	74
5.9	Limites para Disponibilidade e Confiabilidade da Arquitetura A1	76
5.10	Comparação dos limites para Disponibilidade das Arquiteturas A1, A2 e A3	77
5.11	Comparação dos limites para Confiabilidade das Arquiteturas A1, A2 e A3	77

LISTA DE TABELAS

1.1	Comparativo entre as Ferramentas para Dependabilidade	8
4.1	Resultado da análise de importância para confiabilidade	56
4.2	Resultado da análise de importância para disponibilidade	58
4.3	Valores dos índices calculados para o modelo da Figura 4.13	59
4.4	Valores do índice <i>RI</i> calculados para o modelo da Figura 4.14	60
5.1	Índice de importância para confiabilidade para as arquiteturas A1 a A7 .	68
5.2	Valores para disponibilidade, confiabilidade (8760 horas) e custo das Arquiteturas de A1 a A7	68
5.3	Índice de Importância de Componentes para Arquiteturas da Figura 5.5 .	71
5.4	Valores de Disponibilidade, Confiabilidade e Custo para as Arquiteturas A1 a A6	72
5.5	Parâmetros para as Arquiteturas A1, A2 e A3	75
5.6	Resultado da Avaliação dos Limites Superiores para A1 (a cada 3 passos)	75
5.7	Limites para as arquiteturas A1, A2 e A3	76

CAPÍTULO 1

INTRODUÇÃO

A origem dos *data centers* remete ao início da indústria da computação, quando os primeiros sistemas computacionais, que além de serem complexos de operar e manter, exigiam ambientes especiais para o seu funcionamento, como por exemplo sistema eficiente de refrigeração. Desde então, muitos foram os recursos e métodos criados continuamente para acomodar e organizar os *data centers*, tais como *racks* utilizados para acomodar os equipamentos (servidores), pisos elevados para circulação de ar e bandejas de cabos [Augarten 1984].

Atualmente, um *data center* pode ser entendido como uma instalação que comporta sistemas computacionais e componentes relacionados, tais como sistemas de armazenamento e telecomunicações. Geralmente um *data center* inclui fontes de alimentação AC redundantes ou de *backup*, conexões de comunicação de dados redundantes, sistemas de controle ambiental (por exemplo, ar condicionado, detectores de incêndio) e dispositivos de segurança [Marwah et al., 2010a].

Nos últimos anos têm-se notado o crescimento elevado em número, tamanho e potência dos *data centers*. Isto tem sido influenciado principalmente por paradigmas tais como computação em nuvem [Armbrust et al., 2009][Mihailescu et al., 2011], software como serviço (SaaS) [Mckinley et al., 2006] e uma gama enorme de serviços baseados na internet, além de sites de redes sociais e aplicações multimídia [Marwah et al., 2010a].

A facilidade de uso e de acesso a estes serviços tem criado uma dependência dos serviços baseados na Internet, tornando necessária a disponibilização dos serviços 24 horas por dia, 7 dias por semana, além da necessidade de serem altamente confiáveis, pois um determinado serviço fora do ar, mesmo que por pouco tempo, pode levar a grandes prejuízos, facilmente ultrapassando milhões de dólares [Patterson 2002].

Para que estes requisitos (de disponibilidade, confiabilidade, etc) para serviços da Internet sejam garantidos, altos investimentos são necessários, incluindo equipamentos para fornecer redundância. Normalmente o projeto para infraestruturas que adota os equipamentos mais sofisticados, acaba incorrendo em um custo mais elevado [Patterson 2002]. A depender do contexto, é melhor aplicar redundância utilizando equipamentos mais baratos, o que poderá resultar em maiores resultados (em relação a disponibilidade, por exemplo) que utilizar o dispositivo mais poderoso, que certamente será o mais caro [Avizienis et al., 2001].

Além da preocupação com os requisitos já citados, um *data center* envolve custos altos relacionados à aquisição dos equipamentos, à operação, além das multas por violação dos

níveis de serviço dos contratos (*Service Level Agreement* - SLA's) [Leitner et al., 2010] [Marwah et al., 2009].

Em linhas gerais, o custo total de aquisição é o preço líquido do equipamento, acrescido de outras despesas necessárias para a compra do item até chegar ao ponto de utilização, incluindo os custos de compra do item, transporte, preparação e custos de instalação [Paquette 2004]. Os custos com treinamento e integração de sistemas são considerados custos operacionais, que inclui todos os custos para se manter a estrutura funcionando [Marwah et al., 2010a].

Para relacionar estes custos há uma métrica chamada de TCO (*Total Cost of Ownership*) ou custo total da posse [Marwah et al., 2010a]. TCO é uma estimativa financeira que visa avaliar os custos diretos e indiretos relacionados à compra de todo investimento, tal como softwares e hardwares, além dos gastos inerentes de tais produtos para mantê-los em funcionamento, ou seja, os gastos para que se continue proprietário daquilo que foi adquirido [Feria e Nunn, 2006].

Além do TCO, um outro aspecto importante para determinar o custo de um serviço são as violações de SLA (ou multas por quebra de contrato). Estas multas geralmente são baseadas na qualidade dos serviços oferecidos e, quando aplicadas, representam valores elevados [Li et al., 2010].

É importante no processo de análise das infraestruturas, considerar a questão dos custos, uma vez que, aplicar redundância em um determinado componente, baseado em uma métrica que não vincule os custos, pode representar um investimento alto enquanto que aplicar redundância em outro componente, poderá representar um investimento menor, com resultados satisfatórios, em relação aos requisitos de dependabilidade.

É válido destacar ainda que, com a atual discussão acerca do aquecimento global e do esgotamento dos combustíveis fósseis, além do aumento dos custos de energia, questões de sustentabilidade tornaram-se presentes [Callou et al., 2010]. Isso fez surgir o conceito de *data centers* sustentáveis, que são aqueles construídos usando a menor quantidade dos materiais mais adequados e que utilizam fontes de energia mais adequadas ao longo da sua vida, consumindo a menor quantidade possível [Bash et al., 2008].

Para a análise quantitativa do impacto da sustentabilidade, relacionadas a mudanças nas arquiteturas dos *data centers*, projetistas de infraestruturas ainda sentem carência de ferramentas de apoio. De fato, ao se analisar dois *data centers* com disponibilidades semelhantes e diferentes arquiteturas, o impacto de sustentabilidade pode ser muito diferente, assim como a adição de componentes redundantes diferentes poderão causar impactos distintos.

Métricas importantes para comparar equipamentos são: (i) consumo de energia, (ii) utilização de material, (iii) impacto ambiental e (iv) prejuízos naturais irreversíveis para as próximas gerações, tanto na produção como na fase operacional. Portanto, um determinado equipamento pode ser considerado a melhor opção, mesmo tendo um maior consumo de energia, desde que tenha um menor impacto negativo sobre o futuro.

Como um índice para esta avaliação global, a *exergia* de propriedades termodinâmicas é uma métrica interessante. Exergia é definida como a fração máxima de energia que pode ser, teoricamente, convertida em trabalho útil [Dincer e Rosen, 2007]. Por exemplo, a *exergia* contida em um kJ de petróleo é maior que a *exergia* contida em um kJ de água morna. Portanto, estratégias holísticas para lidar com questões conflitantes, tais como elevadas disponibilidades, altos custos econômicos e ambientais, são fatores primordiais para infraestruturas *data center* sustentáveis [Dincer e Rosen, 2007].

Nestas infraestruturas de *data center*, que geralmente compreendem sistemas complexos, nem sempre é trivial realizar a análise de dependabilidade [Laprie et al., 1992], por isso é fundamental o apoio de ferramentas de software especializadas. Técnicas de modelagem, com uma forte base matemática, como *Stochastic Petri Nets* (SPN)[Molloy 1982] [Ajmone Marsan et al., 1984] [Florin e Natkin, 1979] e *Reliability Block Diagrams* (RBD) [Rausand e Høyland, 2004] [Kuo e Zuo, 2003] [Stapelberg 2008] podem ser adotadas para avaliar a dependabilidade de tais infraestruturas.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

O forte aumento do tamanho, capacidade e poder de processamento dos servidores, assim como os avanços de novas tecnologias e equipamentos têm causado um aumento na complexidade das infraestruturas dos *data centers*. Empresas que oferecem serviços de *data centers* devem fornecer um serviço que atenda a expectativa de desempenho (confiabilidade, alta disponibilidade e segurança) que os clientes exigem, além de manterem a preocupação com o impacto ambiental e com os custos relacionados.

As organizações têm seus negócios fortemente baseados nas operações de tecnologia da informação (TI), sendo estas operações cruciais para a continuidade dos negócios. As empresas confiam fortemente em seus sistemas de informação para executar suas operações, ou seja, se um sistema ficar indisponível, as operações da empresa podem ser prejudicadas ou até serem interrompidas completamente. Desse modo, é necessário fornecer uma infraestrutura confiável para as operações de TI, de modo a minimizar qualquer possibilidade de falha.

Para tanto, a análise de dependabilidade é uma atividade essencial que visa fornecer meios para que seja possível promover melhoria da qualidade dos serviços prestados, além de possibilitar o planejamento de melhores infraestruturas para *data centers*. Geralmente a avaliação de dependabilidade tem por objetivo melhorar os níveis disponibilidade e confiabilidade dos serviços, podendo também fornecer informações sobre meios para minimizar os custos e o impacto ambiental (sustentabilidade). Para isto, várias são as técnicas de modelagem que fornecem suporte para análise de dependabilidade. Estas técnicas podem representar os sistemas nos mais diferentes níveis de abstração, sendo adotado aquele mais adequado para a resolução do problema específico que se deseja resolver.

Análise de dependabilidade de um sistema compreende uma gama de métricas que incluem, além de confiabilidade e disponibilidade, o cálculo de outros índices como tempo

médio para falha (*Mean Time To Failure* - MTTF), tempo médio entre falhas (*Mean Time Between Failures* - MTBF), tempo médio para reparo (*Mean Time To Repair* - MTTR) e medidas de importância de componentes (*Component Importance*) [Chang et al., 2004].

No processo de análise da infraestrutura, deve ser possível ao projetista, analisar, juntamente com as métricas de dependabilidade, questões relacionadas a custos. Isto permitirá encontrar qual o melhor arranjo para a infraestrutura, focando nos componentes mais indicados para redundância, para melhoria ou substituição, que possuam custos moderados. Com base nestas informações, além de garantir, no projeto, que o sistema tenha os requisitos de dependabilidade desejados, é possível minimizar os custos da infraestrutura do *data center*.

Dessa forma, os instrumentos de avaliação da confiabilidade devem fornecer uma gama de diferentes técnicas de avaliação que são escolhidas de forma adequada para cada problema. Como mencionado anteriormente, uma forma de aumentar a disponibilidade de sistemas é através da adição de redundância. No entanto, uma vez que a redundância leva a utilização adicional de recursos e energia, seu uso geralmente deverá ter um impacto negativo sobre a sustentabilidade.

Assim é necessário, além da análise das métricas tradicionais de dependabilidade, uma análise de custos e de impacto na sustentabilidade para determinar uma arquitetura custo-efetiva para a construção de infraestruturas *data center*. Esta infraestrutura deve garantir que: (1) os requisitos de energia sejam quantitativamente e qualitativamente atendidos e (2) os requisitos acertados nos SLA sejam cumpridos. Como dito anteriormente, estas infraestruturas geralmente são complexas, por isto é fundamental ferramentas que forneçam suporte a este processo de avaliação.

1.2 OBJETIVOS

O presente trabalho propõe meios que auxiliem no processo de análise de infraestruturas de *data centers*. Estes meios visam aliar análise de dependabilidade e custos, destacando importância de componentes, principalmente quando vinculados ao processo de redundância. Com isto, deseja-se obter estruturas que atendam claramente aos requisitos de dependabilidade minimizando custos de implantação e operação.

A partir das infraestruturas que se deseja analisar, são gerados modelos para análise de dependabilidade. Estes modelos podem ser analisados e dimensionados, para atender aos requisitos de interesse. Ao se obter estes requisitos, os modelos poderão ser utilizados nos projetos de implantação, reestruturação ou redimensionamento das infraestruturas de *data center*.

Estão dispostos, de forma detalhada, os objetivos alcançados durante a realização da pesquisa, que compreendem a realização do objetivo geral:

- definir índices que relacionem custos e dependabilidade, inclusive relacionadas a importância de componentes.

- disponibilizar os novos índices definidos no ambiente RBD das ferramentas ASTRO e Mercury.
- implementar método para obtenção de valores aproximados para modelos RBD.
- implementar método que permita a avaliação de modelos RBD que contenham componentes repetidos.
- obter modelos de infraestruturas de *data center* conforme o modo de operação e falha de cada uma.
- avaliar as infraestruturas a fim de comparar as alternativas baseado nas métricas de dependabilidade e custos.

1.3 TRABALHOS RELACIONADOS

Pesquisas recentes tem abordado temas relacionados aos estudados nesta pesquisa, tanto envolvendo análise de importância de componentes, quanto ferramentas para análise de dependabilidade. Alguns destes trabalhos são descritos nas seções seguintes.

1.3.1 Análise de Importância de Componentes

Alguns trabalhos recentes tem utilizado e estendido índices de importância de componentes tradicionais, além de propor novos índices, adaptando-os para casos específicos.

Si et. al. em [Si et al., 2010], apresenta uma medida para sistemas coerentes binários, que integra a confiabilidade, estrutura e causalidade para descrever as regras dos componentes no sistema. Esta medida integra as medidas de importância para confiabilidade, importância crítica, importância estrutural e importância de Fussel-Vesley e representa o valor esperado da confiabilidade do sistema, baseada na confiabilidade do componente atual.

Em [Zhang et al., 2006], os autores estabelecem índices para confiabilidade para identificar componentes críticos em sistemas de proteção totalmente digitais (*all-digital protection system*), utilizando importância para a confiabilidade e importância crítica. Neste caso, a medida de importância para a confiabilidade é considerada para identificar os pontos fracos que requerem melhorias. Adicionalmente, importância crítica é aplicada para guiar as prioridades nas atividades de manutenção.

Uma comparação entre confiabilidade e custos entre dois *data centers* é apresentada por Wiboonrat [Wiboonrat 2008a] com o intuito de propor uma relação otimizada entre a confiabilidade do sistema e o custo de investimentos para cada caso. São investigadas disponibilidade/confiabilidade dos *data centers* em termos da frequência e duração das ocorrências de falta de energia para calcular ou prever o custo do serviço fora comparado ao custo para melhorar a arquitetura, sendo apresentada uma comparação entre os custo para arquiteturas *Tier III* e *Tier IV*.

[Wiboonrat 2008b] propõe uma anatomia de risco para detectar um ponto único de falha nos sistemas de distribuição de energia dos *data center*, bem como um mecanismo para indicar como melhorar os investimentos com níveis de riscos aceitáveis. Em outras palavras, o trabalho propõe uma técnica para identificar os gargalos do sistema de energia e explica como adotar melhorias nesse sistema de forma ótima, considerando um determinado nível de risco.

Marwah et al. em [Marwah et al., 2010b], apresenta uma abordagem para estimar o impacto da sustentabilidade bem como da disponibilidade em arquiteturas de *data center*. Além disso, cinco arquiteturas de potência de *data center* com diferentes níveis de redundância foram analisadas. A abordagem adotada não considera qualquer meio de suporte aos projetistas sobre quais equipamentos teriam maiores impactos na confiabilidade ou na disponibilidade, quando consideraram as diferentes arquiteturas redundantes.

Alguns dos trabalhos apresentados utilizam ou propõem índices de importância, no entanto não relacionam custos a estes índices. Outros trabalhos tratam de *data center*, propondo melhorias das infraestruturas ou calculando custos durante a operação, mas não relacionam importância e custos na fase de projeto de novas infraestruturas. Diferente destes trabalhos, as métricas de importância propostas nesta pesquisa visam, além de identificar quais os componentes mais relevantes para a confiabilidade ou disponibilidade, relaciona custos, avaliando infraestruturas de *data center*, com o intuito de fornecer suporte ao projetista na fase de projeto, apontando infraestruturas onde os resultados de dependabilidade serão satisfatórios e os investimentos serão moderados.

1.3.2 Ferramentas para Análise de Dependabilidade

Dada a importância e preocupação com o tema, existem algumas ferramentas acadêmicas e comerciais para avaliação de dependabilidade de sistemas em geral. Dentre elas, destacam-se as ferramentas Sharpe [Sahner et al., 1996a] [Sahner e Trivedi, 1987], BlockSim [ReliaSoft 2010], TimeNet [German et al., 1995], GreatSPN [Chiola et al., 1995] e SPNP [Ciardo et al., 1989].

A ferramenta BlockSim [ReliaSoft 2010] fornece uma plataforma para a análise da confiabilidade, manutenibilidade e disponibilidade de sistemas, utilizando os formalismos RBD e *Fault Tree* (FT). O BlockSim também permite simulações de sistemas reparáveis. São utilizados cálculos algébricos para obtenção da confiabilidade. O BlockSim fornece também um sofisticado simulador de eventos discretos para analisar a confiabilidade, manutenibilidade, disponibilidade, capacidade máxima (*throughput*), custo do ciclo de vida (*life cycle cost*) e outras análises relacionadas.

Apesar do BlockSim permitir modelagem por meio de RBD e FT, esta ferramenta não utiliza modelos baseados em estados, dificultando a representação de sistemas que possuam algum comportamento considerado dinâmico, como por exemplo, dependência.

Com a ferramenta Sharpe (*Symbolic Hierarchical Automated Reliability and Performance Evaluator*) [Sahner et al., 1996b] é possível analisar a dependabilidade de sistemas

por meio de modelos RBD, SPN ou MC. Nesta ferramenta as métricas de dependabilidade são obtidas apenas por meio de análise numérica [Balbo e Chiola, 1989]. A ferramenta Sharpe suporta sete tipos de modelos: (1) Diagramas de bloco de confiabilidade série/paralelos (RBD); (2) Árvores de falhas sem nós repetidos; (3) Cadeias de Markov acíclicas; (4) Cadeias de Markov cíclicas irredutíveis; (5) Cadeias de Markov cíclicas com estados absorventes; (6) Cadeias semi-Markov acíclicas; (7) Gráficos acíclicos direcionados série/paralelas [Sahner e Trivedi, 1987].

A ferramenta TimeNet (*Toolkit for Evaluating Non-Markovian Stochastic Petri Nets*) [German et al., 1995] é uma ferramenta para avaliação de Redes de Petri estocásticas. Apresenta um ambiente para avaliação de desempenho e dependabilidade através de modelos SPN, porém não fornece a possibilidade de utilização de modelos combinacionais, como RBD. Os modelos podem ser especificados usando uma interface gráfica de alto nível e componentes especializados são fornecidos para o modelo de avaliação com análise estacionária, de aproximação, e simulação de componentes para Redes de Petri markovianas e não-markovianas [German et al., 1995] e [Vismari].

GreatSPN (*GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets*) [Chiola et al., 1995] é uma outra ferramenta que, assim como a TimeNet apresenta um ambiente para avaliação de desempenho e dependabilidade. Esta ferramenta suporta avaliações utilizando modelos baseados em cadeias de Markov e simulação SPN. No entanto não suporta modelos combinatoriais como RBD. Além disso, esta ferramenta não aceita a definição de estruturas de *reward* em sua análise, permitindo apenas a modelagem de sistemas por meio de Redes de Petri estocásticas e coloridas [Chiola et al., 1995] e [Vismari].

Por fim, a ferramenta SPNP (*Stochastic Petri Net Package*) desenvolvida na Universidade de Duke (Estados Unidos), objetiva a modelagem de solução de Redes de Petri temporizadas e estocásticas, fluidas e estocásticas e modelos markovianas e não-markovianos. Os métodos de solução podem ser divididos em duas categorias principais: métodos numéricos e métodos de simulação [Ciardo et al., 1989] e [Hirel et al., 2000].

Todas as ferramentas aqui apresentadas fornecem suporte para análise de dependabilidade, algumas por simulação, outras por métodos numéricos ou analíticos. Com isso, o ambiente da ferramenta ASTRO visa permitir, além da análise das métricas de dependabilidade convencionais, análise de sustentabilidade, utilizando visão de alto nível de *data center*. A ferramenta permite modelagem multi-modelos (com suporte a RBD e SPN). O ambiente RBD permite, adicionalmente, a análise de importância de componentes, cálculo de métricas de dependabilidade aproximadas e a representação de modelos RBD com blocos repetidos.

A Tabela 1.1 apresenta um comparativo das funcionalidades suportadas por cada uma destas ferramentas e da ferramenta ASTRO, agora estendida para utilização mais ampla.

Tabela 1.1 Comparativo entre as Ferramentas para Dependabilidade

	ASTRO	Sharpe	SPNP	BlockSim	TimeNet	GreatSPN
Modelagem RBD e SPN	x	x				
<i>Closed Forms</i> RBD	x	x	x	x		
Valores Aproximados RBD	x					
Simulação SPN	x				x	x
Avaliação Numérica		x				
Interface Alto Nível	x	x	x	x	x	x
Sustentabilidade	x					
Índices clássicos de importância	x			x		
Índices de importância e custos	x					

1.4 ESTRUTURA DA DISSERTAÇÃO

O Capítulo 1 representa uma introdução à pesquisa, apresentando o contexto em que foi desenvolvido o trabalho, bem como motivação e justificativa para realização da pesquisa. Além disso, são apresentados os objetivos e trabalhos recentes relacionados à presente pesquisa.

O Capítulo 2 apresenta os fundamentos para melhor entendimento do trabalho, destacando os aspectos básicos sobre dependabilidade, com conceitos e definições relacionadas. Na sequência são apresentados modelos gerais para análise de dependabilidade, detalhando Diagramas de Bloco para Confiabilidade, adotado neste trabalho. Por fim são apresentados os métodos de avaliação que foram utilizados, além de conceitos relacionados a importância de componentes em modelos RBD.

O Capítulo 3 descreve a ferramenta Astro. Neste capítulo, são apresentadas a descrição da sua arquitetura e as funcionalidades gerais disponibilizadas para o cálculo de dependabilidade. É dado destaque às novas funcionalidades que foram adicionadas como resultado desta pesquisa.

O Capítulo 4 apresenta os métodos de avaliação desenvolvidos e adicionados na ferramenta descrita no Capítulo 3.

O Capítulo 5 inicia com a apresentação da metodologia para avaliação de importância de componentes. Em seguida são apresentados os estudos de caso realizados em infraestruturas de *data center*, mostrando os resultados obtidos através da ferramenta e onde foram adotados os conceitos aqui apresentados.

Finalmente, o Capítulo 6 apresenta as conclusões resultantes do desenvolvimento desta pesquisa, assim como as principais contribuições para a área de estudo, as limitações, além de propostas para trabalhos futuros.

CAPÍTULO 2

FUNDAMENTOS

Neste capítulo serão apresentados os conceitos básicos de *data center*, incluindo as infraestruturas que o compõem. Também serão introduzidos aspectos básicos de dependabilidade, incluindo seus atributos e informações relacionados. Além disso, será dada uma visão geral de alguns tipos de modelos para avaliação de dependabilidade, sendo expandido o conteúdo relacionado a Diagramas de Bloco para Confiabilidade. Por último serão descritos os métodos de avaliação para obtenção de métricas exatas e aproximadas, baseadas nos modelos apresentados. Todos estes conceitos são fundamentais para o entendimento do formalismo adotado e da pesquisa realizada.

2.1 INFRAESTRUTURAS DE DATA CENTER

Data center (ou centro de dados) é o nome dado ao ambiente especializado onde são concentrados os equipamentos responsáveis pelo processamento e armazenamento dos dados de uma empresa ou organização [Alger 2005].

Um *data center* abriga recursos computacionais importantes em ambientes controlados e sob gestão centralizada, permitindo às empresas operar de acordo com suas necessidades de negócio [Arregoces e Portolani, 2003]. Estes recursos computacionais incluem *mainframes*, servidores de aplicação e web, servidores de arquivos e impressoras, aplicações de software e os sistemas operacionais que o rodam, subsistemas de armazenamento e toda a infraestrutura de rede [Arregoces e Portolani, 2003].

Em linhas gerais, para fins de representação e modelagem, um sistema de *data center* pode ser subdividido em três sub-sistemas: (A) infraestrutura de TI, (B) infraestrutura de potência e (C) infraestrutura de refrigeração [Callou et al., 2010], apresentada na Figura 2.1.

A infraestrutura de TI é responsável pelo processamento e armazenamento das informações. Esta infraestrutura consiste basicamente de três principais componentes - servidores, equipamentos de rede e dispositivos de armazenamento [Marwah et al., 2010a]. Softwares para virtualização de servidores podem ser adotados para compartilhamento de hardware, migração e balanceamento de carga [Veras 2011]. Os dispositivos de armazenamento (por exemplo, conjunto de discos) são normalmente conectados através da rede por uma *Storage Area Network* (SAN), mas podem também se conectar a sistemas de arquivos remotos em *Network Attached Storage* (NAS) através da Ethernet [EMC e Lastemc, 2010] [Alger 2005] [Marwah et al., 2010a].

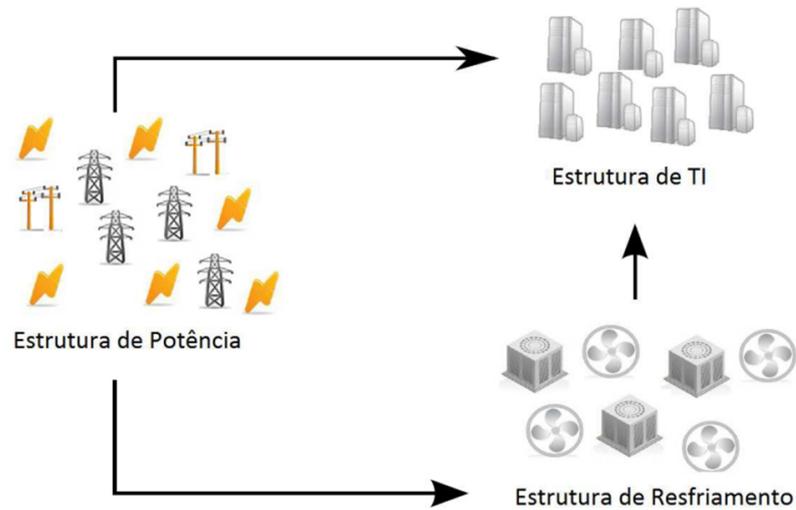


Figura 2.1 Representação de um *Data Center* por subsistemas

O resfriamento de um *data center* é normalmente realizado por meio de um sistema de distribuição de água gelada [Alger 2005]. Este sistema envolve, principalmente, os seguintes equipamentos - manipuladores de ar, *chillers* e torres de resfriamento [Alger 2005] e unidades CRAC (*Computer Room Air Conditioning*) [Callou et al., 2010]. Os manipuladores de ar fazem o ar circular na sala, fazendo com que o calor dissipado dos equipamentos de TI sejam extraídos pelas unidades CRAC e transferido para os sistemas de distribuição de água gelada [Alger 2005]. Os *chillers* extraem o calor do sistema de água gelada e o libera no ambiente através das torres de resfriamento [Marwah et al., 2010a].

Um sistema típico de resfriamento de ambiente de um *data center* possui distribuição de ar frio abaixo do piso ou sob um piso elevado [Bautista e Sharma, 2007]. Este ar frio entra no data center através de painéis perfurados localizados em regiões elevadas do piso, para as entradas dos *racks* [Patel et al., 2001].

Normalmente os racks são dispostos em fileiras separados entre corredores quentes e frios [Bautista e Sharma, 2007]. Os corredores frios fornecem ar gelado para o sistema enquanto os corredores quentes recebem o ar quente dos sistemas computacionais (racks) [Sharma et al., 2004]. O consumo de energia de um sistema de resfriamento de *data center* envolve o trabalho para distribuir o ar frio e para extrair o calor liberado no ambiente [Marwah et al., 2009].

A infraestrutura de potência é responsável por prover energia de forma ininterrupta e adequada, na frequência e voltagem corretas para os equipamentos de TI contidos nos racks e para a estrutura de resfriamento [Marwah et al., 2010a] [Callou et al., 2010]. Basicamente, a infraestrutura de potência consiste de UPS (*Uninterruptible Power Supply*), Transformadores, *Switches* de Transferência (chaveamento), Subpainéis de Distribuição

de Energia e Caixas de Junção [Alger 2005].

Os equipamentos de TI necessitam que a energia seja fornecida de forma ininterrupta. Além disso, é necessário ainda que a energia que alimenta estes equipamentos seja fornecida na corrente e potência corretas, diferentemente da energia fornecida para iluminação e outros equipamentos não essenciais em que estes fatores não representam uma grande preocupação.

Em caso de queda da rede elétrica, um UPS pode fornecer energia sob boas condições [Marwah et al., 2009] ou, alternativamente, a energia pode ser gerada localmente, por meio de geradores [Alger 2005], células de energia ou painéis solares. Cada uma destas fontes possui características distintas referentes à confiabilidade e custo [Marwah et al., 2010a] que precisam ser analisadas para que possam ser implementadas.

2.2 ASPECTOS BÁSICOS SOBRE DEPENDABILIDADE

Devido a expansão dos serviços oferecidos pela Internet, a dependabilidade tem se tornado um atributo de grande interesse no desenvolvimento de *software* e *hardware*, na implantação e operação dos serviços [Maciel et al., 2010].

Os conceitos de dependabilidade foram inicialmente definidos com a publicação do livro *Dependability Basic Concepts and Terminology* [Laprie et al., 1992], e desde então têm sido aplicados e utilizados largamente. Os sistemas que são analisados sob as métricas de dependabilidade são denominados dependáveis [Avizienis et al., 2001].

Dependabilidade de um sistema pode ser definida como a habilidade que ele tem de entregar um determinado serviço no qual se possa justificadamente atestar (provar) [Kuo e Zuo, 2003] [Laprie et al., 1992] [Avizienis et al., 2001]. A entrega de um serviço por um sistema é o modo como ele é percebido por seus usuários. Um usuário é um outro sistema (físico ou humano) que interage com o primeiro sistema por meio de uma interface de serviço. A função, ou especificação funcional, de um sistema é aquilo para o qual ele foi criado. Serviço correto é obtido quando o serviço implementa a função do sistema. A falha do sistema representa o evento que ocorre quando a entrega do serviço não ocorre corretamente [Laprie et al., 1992]. A tradução linear dos termos em inglês *fault* e *failure* (ambas podem ser traduzidas para falha) podem não deixar claro o que significam em algum contexto. Devido a isto, neste trabalho serão adotados os termos ‘falha’ para descrever *failure* e ‘defeito’ para descrever *fault*.

A Federação Internacional para Processamento de Informação (*International Federation for Information Processing - IFIP*) define dependabilidade como “fidelidade” de um sistema computacional que permite o recurso a ser colocado em razão dos serviços que oferece [Basili et al., 2004].

Dependabilidade envolve várias métricas e tem sido utilizada pela comunidade de computação segura e tolerante a falha em vários momentos [Laprie et al., 1992].

Conforme citado nos trabalhos de Avizienis et. al. [Avizienis et al., 2001] e Laprie et.

al. [Laprie et al., 1992], uma exposição sistemática dos conceitos de dependabilidade consistem em três partes: as ameaças, os atributos e os meios pelos quais a dependabilidade é atingida, como apresentado na Figura 2.2.

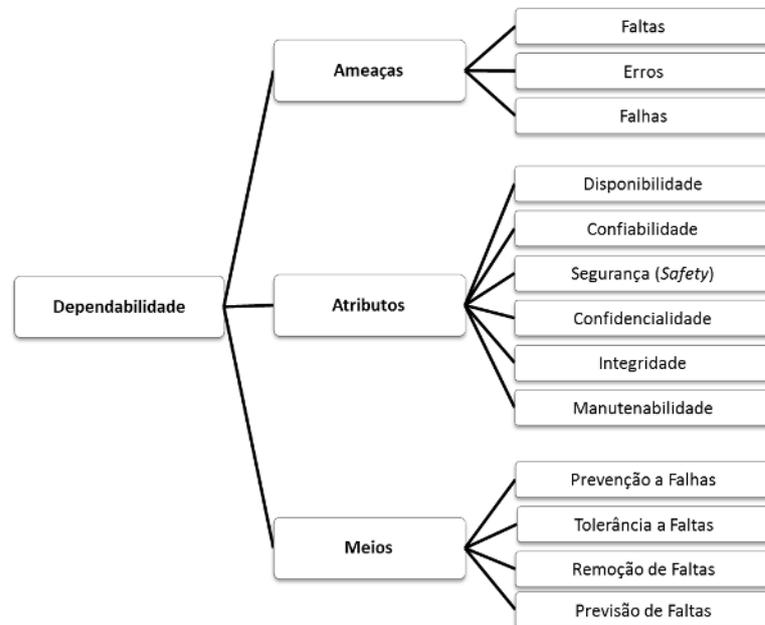


Figura 2.2 Árvore da Dependabilidade. Adaptado de [Avizienis et al., 2001]

A análise de dependabilidade tem se tornado muito frequente pois, cada vez mais, indivíduos e organizações estão desenvolvendo ou adquirindo sistemas computacionais complexos, cujos serviços exigem muita confiança - serviços bancários, equipamentos de controle de satélites, um avião, uma usina nuclear, ou um aparelho de radioterapia, ou para manter a confidencialidade de uma base de dados sensíveis armazenada em grandes *data centers*, levando a uma preocupação maior com questões relacionadas aos sistemas computacionais.

Segundo [Avizienis et al., 2001] os sistemas computacionais são caracterizados por cinco propriedades fundamentais: funcionalidade, usabilidade, desempenho, custos e dependabilidade.

Muitos dos sistemas atuais, *hardware* e *software*, são grandes e complexos e possuem estruturas e características específicas. Para melhorar estes sistemas, é necessário obter, além de sua confiabilidade, outros atributos da dependabilidade. Um sistema pode ser entendido como um conjunto de componentes ou subsistemas conectados, que interagem, de acordo com um projeto específico [Avizienis et al., 2001].

Em um sistema, os tipos de componentes, a quantidade e a qualidade destes e a maneira que eles estão dispostos, afetam diretamente a sua dependabilidade. Desse modo, a principal meta para o estudo de dependabilidade é encontrar o modelo que representa os modos de falha do sistema inteiro baseado nos componentes ou subsistemas dos quais o sistema é composto [Corporation 2003].

2.2.1 Atributos da Dependabilidade

O objeto da análise sobre as questões de dependabilidade está em seus atributos que compreendem: disponibilidade, confiabilidade, segurança, confidencialidade, integridade e manutenibilidade [Avizienis et al., 2001]. Conforme o interesse de aplicação do sistema, ênfase diferente pode ser dada a diferentes atributos. Dependendo da função e complexidade do sistema, algumas métricas podem ser melhor utilizadas que outras.

A confiabilidade de um item ou sistema é a probabilidade de que um sistema desempenhará satisfatoriamente (ou adequadamente) o seu propósito específico por um dado período até a ocorrência da primeira falha [Maciel et al., 2010] [Xie et al., 2004] [Leemis 1995]. Ou seja, é a probabilidade de que o sistema não falhará até o tempo especificado [Rausand e Høyland, 2004] [Kuo e Zuo, 2003].

Por exemplo, uma confiabilidade de 95% para 8760 horas, significa que há 95% de chances de o sistema não ter falhado em 8760 horas de operação.

Para descrever a confiabilidade de um dado sistema, é necessário conhecer a sua configuração, o estado em que ele é definido como operacional e as regras de operação [Kuo e Zuo, 2003]. Matematicamente, a função de confiabilidade $R(t)$ é a probabilidade de que o sistema irá funcionar corretamente sem falha no intervalo de tempo de 0 a t [Maciel et al., 2010], conforme apresentado na Equação 2.1.

$$R(t) = P(T > t), t \geq 0 \quad (2.1)$$

onde T é uma variável aleatória representando o tempo de falha ou tempo para falha.

A probabilidade de falha, ou inverso da confiabilidade, é então representada por:

$$F(t) = 1 - R(t) = P(T \leq t) \quad (2.2)$$

que é conhecida como a função de distribuição de T .

Normalmente deseja-se conhecer o tempo esperado para que ocorra a próxima falha. Esta métrica é chamada de Tempo Médio Para Falha (*Mean Time To Failure - MTTF*) e é definida como o valor esperado de operação do sistema antes da falha ocorrer [Xie et al., 2004].

Quando um sistema para de funcionar, normalmente aplicam-se operações de reparo para corrigir a falha. O sistema então recupera o seu estado operacional em virtude de ajustes feitos ou mesmo de substituição de componentes [Xie et al., 2004].

Partindo deste perspectiva, manutenibilidade é definida como a probabilidade que um sistema falho irá ser restaurado para o estado operacional por um dado período de tempo quando a manutenção é realizada de acordo com procedimentos estabelecidos e os recursos [Dhillon 2002] [Xie et al., 2004].

Geralmente, manutenibilidade é a probabilidade de se isolar e reparar uma falha em um sistema dentro de um determinado momento. O pessoal de manutenção deve trabalhar

com projetistas de sistema para garantir que o produto do sistema possa manter o custo-benefício.

Uma medida importante, relacionada a manutenibilidade, é o Tempo Médio Para Reparo (*Mean Time To Repair* - MTTR).

Outro atributo de dependabilidade é a disponibilidade, que pode ser instantânea ou estacionária.

A disponibilidade instantânea representa a probabilidade de que o sistema não falhe em um determinado tempo [Xie et al., 2004], diferentemente da confiabilidade que se refere ao período de tempo em que o sistema estará livre de falhas. A disponibilidade instantânea também pode ser definida como a fração de tempo que o sistema provê serviços a seus usuários [Kuo e Zuo, 2003] ou a prontidão do sistema para a entrega do serviço correto em um dado momento [Stapelberg 2008] [Avizienis et al., 2001].

Matematicamente, a disponibilidade instantânea pode ser representada por:

$$A(t) = Pr(\text{Sistema estará disponível ou operacional no tempo } t)$$

A função de disponibilidade, que é uma função complexa do tempo, tem um estado estacionário simples ou expressão assintótica. De fato, geralmente há preocupação principalmente com sistemas que executam continuamente por um longo período de tempo [Xie et al., 2004].

Assim, a disponibilidade em estado estacionário [Stapelberg 2008] pode ser entendida como a porcentagem de tempo em que um componente ou sistema pode desempenhar sua função [Wiboonrat 2008a]. Em outras palavras, é a probabilidade que o sistema estará funcionando em qualquer momento no tempo. Esta medida leva em conta tanto confiabilidade quanto manutenibilidade.

A disponibilidade em estado estacionário ou assintótica é dada por:

$$A = \lim_{t \rightarrow \infty} A(t) = \frac{\text{SystemUpTime}}{\text{SystemUpTime} + \text{SystemDowntime}} = \frac{MTTF}{MTTF + MTTR} \quad (2.3)$$

A disponibilidade de equipamentos ou subsistemas pode ser melhorada usando diferentes níveis de redundância de equipamentos [Semaan 2008].

O tempo médio entre falhas (*Mean Time Between Failures* - MTBF) é uma outra medida importante em sistemas reparáveis. Significa que o sistema falhou e foi reparado. Assim como MTTF e MTTR, MTBF é o valor esperado para uma variável aleatória do tempo entre falhas. $MTBF = MTTR + MTTF$.

2.3 MODELOS PARA AVALIAÇÃO DE DEPENDABILIDADE

Várias são as abordagens existentes e utilizadas para representar e analisar a dependabilidade de sistemas. Estes modelos podem ser classificados em combinatoriais e modelos

baseados em estados [Maciel et al., 2010]. Modelos baseados em estados podem também ser chamados de não-combinatoriais, e os modelos não baseado em estados são chamados de combinatoriais.

Modelos combinatoriais capturam as condições que levam o sistema a falhar (ou a operação) em termo da relação estrutural entre seus componentes. Já os modelos baseados em estados representam o comportamento do sistema (atividades de falha e de reparo) pelos seus estados e pela ocorrência dos eventos [Maciel et al., 2010].

Os formalismos Diagramas de Blocos para Confiabilidade (RBD) [Rausand e Høyland, 2004], Grafos para a Confiabilidade (*Reliability Graph* - RG) [Sahner et al., 1996a] e Árvores de Falta (*Fault Tree* - FT) [Vesely e Roberts, 1987] são os modelos combinatoriais mais largamente utilizados na análise de confiabilidade [Maciel et al., 2010]. Entre os modelos baseados em estados destacam-se os modelos de Markov e derivados e Redes de Petri.

Os formalismos de RBD, RG e FT são definidos sobre uma premissa: os componentes precisam ser estatisticamente independentes. Eles não permitem a modelagem de comportamentos conhecidos como dinâmicos, tais como compartilhamento de carga.

2.3.1 Diagramas de Blocos para a Confiabilidade

Diagramas de Blocos para Confiabilidade são redes de blocos funcionais conectados de acordo com o efeito de cada falha de bloco na confiabilidade do sistema [Maciel et al., 2010], sendo uma das ferramentas convencionais mais comuns para análise de confiabilidade de sistemas [Maciel et al., 2010]. Uma das vantagens de usar o RBD é a facilidade de expressar e avaliar a confiabilidade e disponibilidade [Xie et al., 2004].

Por ser um modelo combinacional, é possível calcular disponibilidade e confiabilidade por meio de fórmulas fechadas (*closed forms*). Estas fórmulas tornam a obtenção do resultado mais rápida se comparado à simulação, por exemplo.

RBD foi inicialmente proposto como um modelo para calcular a confiabilidade em sistemas grandes e complexos usando diagramas de blocos para representar os relacionamentos de falhas [Abd-Allah 1997]. Este formalismo foi estendido e pode ser usado para calcular outras métricas de dependabilidade, tais como disponibilidade, confiabilidade e manutenibilidade.

RBD é uma descrição gráfica da conexão dos componentes do sistema que podem ser usados para determinar o estado geral do sistema a partir do estado dos seus componentes [Sahner et al., 1996b]. Por exemplo, RBDs indicam como o funcionamento dos componentes do sistema afetam o funcionamento do sistema.

A estrutura de um modelo RBD estabelece a interação lógica entre os componentes, definindo quais combinações de funcionamento dos componentes garantem a operação do sistema. Mais especificamente, o sistema é representado por subsistemas ou componentes conectados de acordo com a sua função do relacionamento de confiabilidade [Sahner et al., 1996b]. Uma conexão paralela é utilizada para mostrar redundância de

componentes e ela cria múltiplos caminhos. Quando os componentes estão conectados sem redundância, ou de forma sequencial, eles podem ser representados por uma conexão série.

Um diagrama de blocos para confiabilidade contém uma entrada (ou fonte - no lado esquerdo) e uma saída (ou destino - do lado direito). Entre os nós de entrada e saída, o sistema pode ser constituído por estruturas de blocos em série, paralelo, *bridge* (ponte), blocos *k-out-of-n* (*k de n*), estruturas de redes em geral ou combinações destas estruturas [Kuo e Zuo, 2003]. Estes blocos (normalmente retângulos, mas podem ser também círculos) representam componentes e são conectados por arestas, conforme figura 2.3.

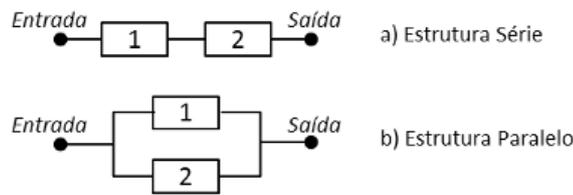


Figura 2.3 Relacionamentos básicos em um Diagrama de Blocos para a Confiabilidade

Em um modelo RBD, o diagrama lógico é organizado para indicar quais combinações de componentes (que quando estão no modo operacional) garantem a operação do sistema.

Nos modelos RBD, se existir pelo menos um caminho conectando os pontos de entrada e saída do modelo o sistema está operacional, ou seja, se blocos pararem de funcionar de modo a interromper a conexão entre os pontos de entrada e saída, o sistema falha [Distefano et al., 2006] [Kuo e Zuo, 2003] [Sathaye et al., 2000]. Este é um modelo orientado ao sucesso [Maciel et al., 2010].

Os modelos mais simples e comuns dão suporte somente a estruturas séries e paralelas. Se para o sistema estar funcionando é necessário que todos os componentes estejam operacionais (funcionando), os blocos, nestes modelos, serão conectados em série. Desta forma, uma conexão em série representa uma dependência direta entre a conexão dos componentes. Por outro lado, se para o sistema pode funcionar com apenas um componente operacional, então os blocos estão conectados em paralelo. Em outras palavras, uma conexão em paralelo é usada para representar redundância.

Blocos *k-out-of-n* representam estruturas em que o subsistema pode funcionar se *k* ou mais componentes estão no estado operacional [Xie et al., 2004] [Kuo e Zuo, 2003]. Por exemplo, uma estrutura em que haja cinco componentes e necessita-se de três funcionando para prover o serviço esperado, temos uma estrutura 3-out-of-5 (ou 3 de 5). As estruturas em série e paralela são casos especiais de estruturas *k-out-of-n*, uma estrutura em série é uma *n-out-of-n* e uma estrutura em paralelo é uma estrutura 1-out-of-*n* [Sahner et al., 1996b] [Kuo e Zuo, 2003].

Considere a estrutura composta de componentes independentes como apresentado na Figura 2.3, onde $p_i = P\{x_i = 1\}$ é a função de probabilidades dos blocos b_i . Estas probabilidades podem ser disponibilidade ou confiabilidade por exemplo.

Assim, para a estrutura em série (conforme Figura 2.3a) com n componentes, a confiabilidade do sistema é representada por:

$$R_s(t) = \prod_{i=1}^n R_i(t) \quad (2.4)$$

onde $R_i(t)$ corresponde a confiabilidade do bloco b_i no instante de tempo t .

Similarmente, outras métricas probabilísticas podem ser calculadas para estruturas em série.

Agora, para estruturas em paralelo (veja Figura 2.3b) com n componentes, a confiabilidade do sistemas é:

$$R_p(t) = 1 - \prod_{i=1}^n (1 - R_i(t)) \quad (2.5)$$

onde $R_i(t)$ corresponde à confiabilidade do bloco b_i no instante de tempo t .

Similarmente, outras métricas probabilísticas podem ser calculadas para estruturas em paralelo.

Os blocos em qualquer estrutura em série ou em paralelo podem ser mesclados em um novo bloco com a expressão confiabilidade das equações acima. Usando essas combinações, qualquer sistema série-paralelo pode ser eventualmente transformado em um bloco e sua confiabilidade pode ser facilmente calculada usando repetidamente as equações citadas [Xie et al., 2004].

Há algumas premissas importantes que devem ser consideradas em modelos RBD como:

- a confiabilidade ou disponibilidade de cada bloco individualmente é conhecida ou estimada;
- as falhas dos blocos são estatisticamente independentes;
- os blocos são bimodais, ou seja, possuem apenas dois estados possíveis: ou funcionam ou falham (degradação dos serviços não é suportada) e,
- todos os caminhos existentes são representados.

Caso a premissas anteriores sejam atendidas pelo sistema de interesse, então é possível calcular as métricas de dependabilidade do sistema representando-o através de um modelo RBD.

2.4 FUNÇÃO DE ESTRUTURA E FUNÇÃO LÓGICA

As funções de estrutura e lógica são funções que representam a estrutura e permitem conhecer o estado do sistema através de sua solução. Estas funções são utilizadas para auxiliar a avaliação de dependabilidade dos modelos série e paralelo. Ambas as funções possuem basicamente o mesmo objetivo, fica a critério do projetista adotar uma ou outra.

2.4.1 Função de Estrutura

Esta função representa, na forma de uma expressão matemática, a estrutura do sistema. Ela é utilizada para indicar o relacionamento entre o estado do sistema e o estado dos componentes [Kuo e Zuo, 2003]. A partir do valor esperado da função estrutural é obtida a expressão para cálculo das métricas probabilísticas dos sistemas série/paralelo.

Para obtenção desta função, têm-se que o estado de cada componente corresponde a uma variável aleatória que pode ter apenas dois valores possíveis, conforme a seguir:

$$\phi(x_i) = \begin{cases} 1, & \text{se o componente funciona} \\ 0, & \text{se o componente falha} \end{cases} \quad (2.6)$$

onde $\phi(x_i)$ é a função estrutural do componente e x_i indica o estado do componente i para $1 \leq i \leq n$ e n o número de componentes. Se o componente i funciona, $x_i = 1$ e se ele falha, então $x_i = 0$.

Se os estados de todos os componentes são conhecidos, então o estado do sistema também pode ser obtido [Kuo e Zuo, 2003]. O estado do sistema é uma função determinística dos estados dos componentes. Dessa forma, podemos escrever:

$$\phi = \phi(\mathbf{x}) = \phi(x_1, x_2, \dots, x_n) \quad (2.7)$$

onde $\phi(\mathbf{x})$ é a função de estrutura do sistema. Cada estrutura única de um sistema, corresponde a uma única função de estrutura $\phi(\mathbf{x})$.

Para um sistema em série com n componentes, a função de estrutura é dada por:

$$\phi(\mathbf{x}) = x_1 * x_2 * \dots * x_n \quad (2.8)$$

Para um sistema em paralelo, a função de estrutura pode ser escrita como:

$$\phi(\mathbf{x}) = (1 - (1 - x_1)(1 - x_2)\dots(1 - x_n)) \quad (2.9)$$

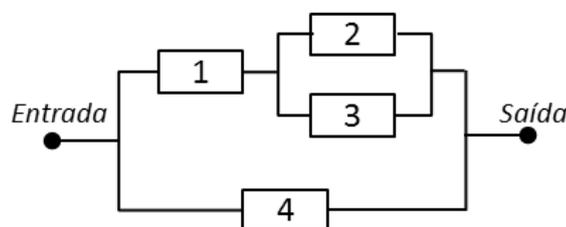


Figura 2.4 Exemplo sistema com construções série e paralelo

Exemplo: O sistema fornecido na Figura 2.4 é composto por estruturas em série e em paralelo. A sua função de estrutura é dada por:

$$\phi(x_1, x_2, x_3, x_4) = 1 - (1 - x_1(1 - (1 - x_2)(1 - x_3)))(1 - x_4) \quad (2.10)$$

É possível notar que a Equação 2.10 representa o estado do sistema como uma função do estado dos seus componentes. Neste caso o sistema estará no estado 1, se os componentes 1 e 2 ou 1 e 3 ou o componente 4 estiverem no estado 1. O sistema estará no estado 0 (zero) se, e somente se, ambos 1 e 4 ou 2, 3 e 4 simultaneamente forem 0.

A função estrutural definida anteriormente é uma função matemática discreta. Ela possui entradas binárias e produz um resultado binário, normalmente 0 ou 1.

2.4.2 Função Lógica

Esta função define o estado do sistema baseado em álgebra booleana e o sistema é representado por expressões lógicas.

Em alguns casos, simplificar a função de estrutura pode não ser uma tarefa fácil. A função lógica pode ser adotada para simplificar as funções de sistemas através de Álgebra Booleana [Maciel et al., 2010].

Notações para a função lógica:

- x_i : o evento em que o componente i funciona, $1 \leq i \leq n$ e n é o número de componentes
- \bar{x}_i o complemento de x_i , indicando que o componente i é falho, $1 \leq i \leq n$
- $S(1, 2, \dots, n)$: o evento em que o sistema com os componentes $\{1, 2, \dots, n\}$ funciona.
- $\bar{S}(1, 2, \dots, n)$: o complemento de $S(1, 2, \dots, n)$, indicando o evento em que o sistema com os componentes $\{1, 2, \dots, n\}$ é falho.

Com estas notações, a função lógica de um sistema em série com n componentes pode ser escrita como:

$$S_{serie} = x_1 x_2 \dots x_n \quad (2.11)$$

E a função lógica de um sistema em paralelo pode ser escrita como:

$$\bar{S}_{paralelo} = \bar{x}_1 \bar{x}_2 \dots \bar{x}_n \quad (2.12)$$

Dessa forma, o sistema representado na Figura 2.4 possui uma função lógica tal como a seguir:

$$\bar{S}(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_1 \bar{x}_3 \bar{x}_4 \quad (2.13)$$

2.5 SISTEMAS COERENTES

Um sistema coerente pode ser definido como um sistema com função de estrutura $\phi(x)$ se, e somente se, $\phi(x)$ é não-decrescente em cada componente x_i para $1 \leq i \leq n$ e cada (todo) componente é relevante.

Um componente é relevante para o sistema se existe pelo menos um vetor de estados \mathbf{x} em que o estado do componente i define o estado do sistema. Em outras palavras, quando outros componentes encontram-se em um certo estado (operacional ou falho), especificado por $(x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$, o valor de $\phi(x_1, x_2, \dots, x_n)$ é igual a x_i . Dadas estas condições, quando o componente i funciona, o sistema funciona; quando o componente i falha, o sistema falha [Kuo e Zuo, 2003].

Outro ponto relevante é que melhorar o desempenho de um componente, normalmente não piora o desempenho do sistema [Kuo e Zuo, 2003]. Por exemplo, a substituição de um componente falho em um sistema funcionando normalmente não faz o sistema falhar. É claro que, substituir um componente falho em um sistema falho não necessariamente recupera o funcionamento do sistema porque podem haver outros componentes falhos no sistema, que o impeçam de funcionar. Normalmente, assume-se que a função de estrutura de todo sistema é uma função não-decrescente do estado de todos os componentes [Kuo e Zuo, 2003].

Desse modo, para um sistema coerente, dado dois vetores, cada qual com n elementos, \mathbf{x} e \mathbf{y} , pode-se escrever $\mathbf{x} < \mathbf{y}$ se $x_i < y_i$ para cada i e $x_i < y_i$ para pelo menos um i ($1 \leq i \leq n$). Em resumo, pode-se dizer que o vetor \mathbf{x} é menor que o vetor \mathbf{y} [Kuo e Zuo, 2003].

Baseado nesta definição, segundo [Kuo e Zuo, 2003], um sistema coerente satisfaz as seguintes condições:

1. $\phi(\mathbf{0}) = 0$, ou seja, o sistema é falho quando todos os componentes são falhos.
2. $\phi(\mathbf{1}) = 1$, ou seja, o sistema funciona quando todos os componentes funcionam.
3. Se $x < y$, então $\phi(x) < \phi(y)$, ou seja, a melhoria de qualquer componente não degrada o desempenho do sistema.
4. Para todos os componentes i , existe um vetor de estados de componentes em que o estado do componente i dita o estado do sistema.

2.6 CAMINHOS E CORTES MÍNIMOS

Os sistemas computacionais são formados por conexões que permitem a sua operação. As funções de estrutura definem o arranjo dos componentes no sistema. Estes arranjos podem ser expressos por conjuntos de caminhos e cortes.

Dado um sistema formado por mais de um componente, se for necessário que todos os componentes funcionem para garantir o funcionamento do sistema, é possível afirmar

que estes componentes constituem o fluxo que permite a operação do sistema. Em outras palavras, existe apenas uma forma dele funcionar com todos os componentes funcionando.

No entanto, se há combinações de componentes que, quando funcionam, garantem o funcionamento do sistema, podemos dizer que há fluxos diversos que garantem a operação do sistema. Por exemplo, considere um sistema de iluminação residencial, onde há 2 lâmpadas para iluminar o mesmo cômodo. Caso uma destas lâmpadas falhe (queime) o sistema (de iluminação) continuará operando, pois a outra lâmpada garantirá a iluminação do ambiente, ou seja, o sistema possui dois fluxos, sendo cada um por uma lâmpada diferente. Estes fluxos representam os caminhos.

De forma similar, um componente, ou conjunto de componentes que quando falham, levam o sistema a sair de operação, representa uma corte. Ou seja, as combinações de componentes que, quando saem de operação simultaneamente, levam o sistema a quebra, são os cortes.

Assim, um caminho é um conjunto de componentes em que o funcionamento simultâneo garante o funcionamento do sistema, enquanto um corte é um conjunto de componentes em que a falha simultânea leva o sistema a falha.

Um caminho mínimo é um conjunto mínimo de componentes em que o funcionamento simultâneo garante o funcionamento do sistema. Da mesma forma, um corte mínimo é um conjunto de componentes em que a falha simultânea dos componentes leva o sistema a falha [Kuo e Zuo, 2003].

Para um sistema em série, cada componente compõe um corte mínimo e o conjunto de todos os componentes é um caminho mínimo. Assim, um sistema em série tem n cortes distintos e somente um caminho mínimo. Para sistemas paralelos, cada componente é um caminho mínimo e todos os componentes foram um corte mínimo. Assim, um sistema paralelo tem n caminhos mínimos e apenas um corte mínimo. Isto pode ser observado na Figura 2.5

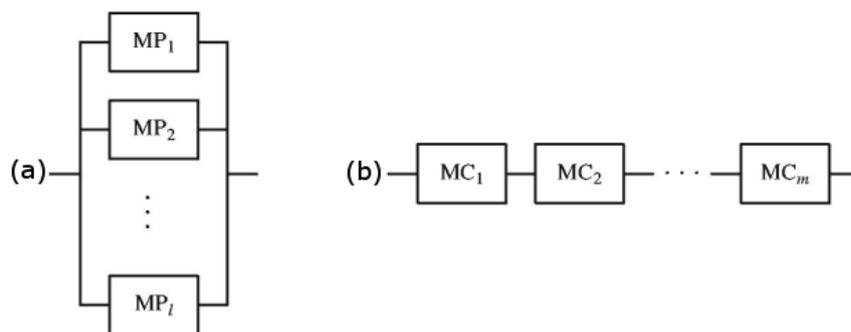


Figura 2.5 Diagrama de Blocos para Confiabilidade de sistemas coerentes com (a) caminhos mínimos e (b) cortes mínimos. Fonte [Kuo e Zuo, 2003]

Para garantir que o sistema irá funcionar, pelo menos um caminho mínimo deve funci-

onar. O sistema pode ser considerado como uma estrutura em paralelo com cada caminho mínimo como um componente. Para o sistema falhar, pelo menos um corte mínimo deve falhar. Como resultado, um sistema pode ser considerado como uma estrutura em série em que cada corte mínimo compreende um componente [Kuo e Zuo, 2003].

Baseado nas definições de caminhos e cortes mínimos, os componentes em cada caminho mínimo compõem uma estrutura em série e os componentes em cada corte mínimo são uma estrutura em paralelo.

Cortes e caminhos mínimo podem ser usados para derivação da função de estrutura de qualquer sistema coerente. No entanto, deve-se notar que caminhos mínimos diferentes podem conter componentes em comum, assim como cortes mínimos, aumentando a complexidade das operações [Kuo e Zuo, 2003].

Considere a Figura 2.6 a qual apresenta um modelo RBD formado por três componentes. Neste modelo pode-se notar que para o sistema funcionar, basta que funcionem os componentes A e B simultaneamente ou então A e C. Desse modo, pode-se afirmar que AB compreende um caminho e AC compreende outro caminho mínimo.

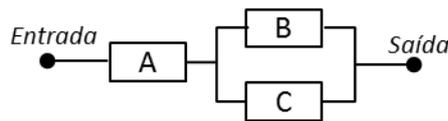


Figura 2.6 Modelo Série/Paralelo

Similarmente, observando a Figura 2.6, é possível perceber que para que o sistema falhe, é necessário que, ou falhe o componente A, ou então falhem, simultaneamente, os componentes B e C. Desse modo, pode-se dizer que A e BC compreendem os dois cortes mínimos do modelo.

Os cortes e caminhos aqui considerados devem ser mínimos, ou seja, apenas a combinação mínima de componentes que levem ao sucesso ou quebra do sistema. Por exemplo, poderíamos dizer que se funcionarem os componentes A B e C do modelo, o sistema, funcionaria, mas este não compreende um caminho mínimo, uma vez que se os 3 componentes estiverem funcionando, a falha de B ou C, separadamente, não impede o funcionamento do sistema por este caminho, não sendo portanto mínimo.

Um outro exemplo pode ser o da estrutura apresentada na Figura 2.4 onde existem 3 caminhos mínimos $\{[x_1x_2], [x_1x_3], [x_4]\}$ e 2 cortes mínimos $\{[x_1x_4], [x_2x_3x_4]\}$.

2.7 MÉTODOS DE AVALIAÇÃO

Vários são os métodos disponíveis para avaliação da dependabilidade de estruturas de sistemas computacionais baseados em composições simples, como série, paralelo e combinações dessas. Nesta seção serão apresentados métodos para avaliação de dependabilidade de modelos combinacionais, uma vez que na pesquisa foi utilizado o modelo RBD.

Com o intuito de simplificar a notação, nas equações apresentadas, as probabilidades dos componentes (referentes a confiabilidade e disponibilidade) serão representados pela variável p_i e a probabilidade do sistema por P .

Normalmente, quando se deseja calcular os valores de dependabilidade, o interesse está em obter um valor exato para todo o sistema. Há casos em que, quando os sistemas são grandes ou complexos, calcular valores exatos para as métricas de dependabilidade exigirá uma considerável quantidade de tempo, além da exigência de altos recursos computacionais.

Nos casos em que não se conta com recursos computacionais necessários ou não é possível aguardar o tempo de simulação para obtenção da métrica, existem algumas técnicas para obtenção de valores aproximados que podem levar menos tempo para serem obtidos e exigirem menos recursos [Kuo e Zuo, 2003]. Nesta seção serão abordados alguns destes métodos aproximados, além dos métodos para obtenção dos valores exatos.

2.7.1 Métodos Exatos

A estratégia mais simples e comum para avaliar a confiabilidade (ou disponibilidade) de um sistema composto por componentes independentes é através de sua respectiva definição [Maciel et al., 2010]. Assim, considere um sistema S e a sua respectiva função de estrutura. A confiabilidade do sistema poderia ser definida como apresentado na Equação 2.14.

$$R_s = P\{\phi(\mathbf{x}) = 1\} = E\{\phi(\mathbf{x})\} \quad (2.14)$$

Resumindo, os passos principais para se avaliar a probabilidade de falha de um sistema, segundo Maciel et.al. [Maciel et al., 2010] adotando o método básico são:

1. obter a função de estrutura do sistema
2. remover os valores de cada variável x_i e
3. substituir cada variável x_i pela probabilidade p_i do respectivo bloco

As estruturas mais fundamentais de sistemas para a confiabilidade são série e paralelo. A dependabilidade destes sistemas são analisadas através das equações apresentadas na Seção 2.3.1.

A Figura 2.7 apresenta um modelo RBD formado de estruturas série e paralelo, contendo 3 componentes.

Para exemplificar a aplicação do método exato para este modelo, inicialmente é obtida a função de estrutura, apresentada na Equação 2.15.

$$\phi(\mathbf{x}) = x_1(1 - (1 - x_2)(1 - x_3)) \quad (2.15)$$

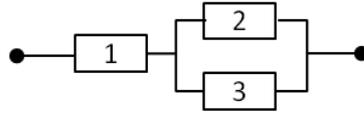


Figura 2.7 Modelo RBD com estruturas série e paralelo

Na Equação 2.15, substitui-se os valores das variáveis x_i (que possuem valores 0 ou 1) pela probabilidade p_i de cada um dos blocos (confiabilidade, por exemplo), obtendo a Equação 2.16. Após, resolvendo a referida expressão, é obtido o valor da métrica de interesse.

$$P_s = p_1(1 - (1 - p_2)(1 - p_3)) \quad (2.16)$$

Outra forma de se obter o valor exato da métrica, é aplicando reduções do modelo RBD. Para ilustrar a aplicação do método, considere um sistema em que o Digrama de Blocos para a Confiabilidade é apresentado na Figura 2.8(a) que é composto por 4 componentes. A confiabilidade e a não-confiabilidade do componente i é representada por p_i e q_i respectivamente. Para obter o valor exato da confiabilidade, é utilizado o procedimento de reduções em série/paralelo.

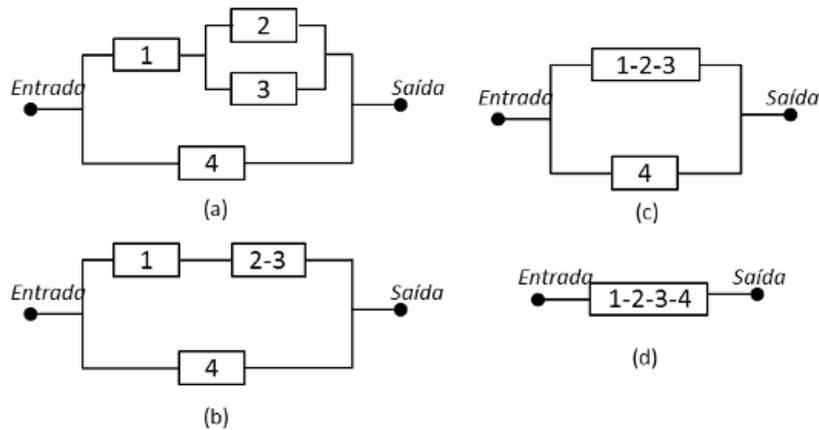


Figura 2.8 Usando reduções série e paralelo no modelo RBD

Para a estrutura dada na Figura 2.8(a), é possível notar que os componentes 2 e 3 formam um subsistema em paralelo, de modo que é possível representá-los por um componente chamado 2-3. O inverso da confiabilidade do componente 2-3 é igual a $q_{2-3} = q_2q_3$.

Após esta redução, observando a Figura 2.8(b), nota-se que os componentes 1 e 2-3 compõem um subsistema em série, sendo possível, então, representá-los através de um componente, que será chamado de 1-2-3. O valor da confiabilidade deste componente será $p_{1-2-3} = p_1(1 - q_{2-3})$. Esta redução gera a Figura 2.8(c).

Nas Figura 2.8(c) o componente 4 e o componente 1-2-3 formam um componente em paralelo que, novamente, podem ser transformados em um único componente (veja Figura 2.8(d)). Aplicando a redução, teremos o inverso da confiabilidade deste componente dada por $q_{1-2-3-4} = q_4 q_{1-2-3} = q_4(1 - p_{1-2-3})$. Assim, a confiabilidade do sistema é igual a:

$$R_s = 1 - (1 - p_4)[1 - p_1(1 - q_2 q_3)]$$

Ao se aplicar, sucessivamente, estas reduções, é possível obter-se um componente simples, cujo valor da confiabilidade (ou métrica de interesse) será o valor exato para aquele dado sistema [Maciel et al., 2010].

Há casos, como ocorre com a estrutura de ponte, em que apenas reduções série/paralelo não contemplam a obtenção do valor exato para a métrica de interesse. Neste caso, podem-se utilizar outras técnicas, como é o caso da **decomposição pivotal** [Kuo e Zuo, 2003] [Maciel et al., 2010].

O método da decomposição pivotal é baseado no conceito de probabilidade condicional [Kuo e Zuo, 2003]. A equação a seguir ilustra a idéia do método:

$$\begin{aligned} & \Pr(\text{Sistema funcionar}) \\ &= \Pr(\text{componente } i \text{ funciona}) \Pr(\text{sistema funciona} \mid \text{componente } i \text{ funciona}) \quad (2.17) \\ &+ \Pr(\text{componente } i \text{ falha}) \Pr(\text{sistema funciona} \mid \text{componente } i \text{ falha}) \end{aligned}$$

Segundo [Kuo e Zuo, 2003], a eficiência deste método depende da facilidade de avaliação da probabilidade condicional. Isto significa que o conjunto de componentes pode ser decomposto desempenhando um papel de suma importância na eficiência do método. Se a decomposição do componente selecionado resulta em dois sistemas onde seja preciso aplicar reduções série e paralelo novamente, a eficiência na avaliação da confiabilidade do sistema não foi alcançada.

O método da decomposição pivotal é baseado na enumeração dos estados do componente selecionado. Uma vez que todos os componentes tem dois estados, uma aplicação do método de decomposição para um sistema com n componentes sujeitos a falhas gera duas diferentes estruturas de sistemas cada uma com $n - 1$ componentes sujeitos a falha. Se a confiabilidade destes dois sistemas menores é conhecida, restam duas multiplicações e uma adição para encontrar a confiabilidade de um sistema com n componentes. Aplicar apenas decomposição pivotal para calcular a confiabilidade de um sistema gera uma função exponencial. No entanto, combinando o método da decomposição pivotal com as reduções série e paralelo e utilizando as expressões de confiabilidade para as estruturas padrão de sistemas produz um resultado muito mais rápido [Kuo e Zuo, 2003].

2.7.2 Métodos Aproximados

Alguns métodos são explorados pela literatura para obtenção dos valores aproximados dos atributos de dependabilidade. Dentre eles destacam-se o método da Inclusão/Exclusão,

Min-Max Bounds, Esary-Proschan que provêm sucessivos valores de limites superiores e inferiores, baseados nos caminhos e cortes [Kuo e Zuo, 2003].

Neste trabalho será dado destaque ao método Soma dos Produtos Disjuntos, onde todos os conjuntos (caminhos ou cortes) são incluídos, diferentemente de outros métodos onde nem todas as iterações geram valores válidos.

2.7.2.1 Método Soma dos Produtos Disjuntos O método Soma dos Produtos Disjuntos (SPD) usa os caminhos mínimos ou os cortes mínimos para avaliar a probabilidade de falha ou de operação do sistema avaliando a probabilidade de união dos caminhos (ou cortes). A probabilidade de união dos caminhos mínimos podem ser expressos através da função lógica do sistema. A função lógica pode ser expressa como a união de vários termos. Se estes termos da função lógica forem disjuntos, então existe uma correspondência de um-para-um entre esta expressão da função lógica do sistema e a medida da confiabilidade do sistema. Assim, o foco do método é expressar a função lógica do sistema como a união dos termos disjuntos. Cada um destes termos disjuntos é um produto dos eventos em que os componentes individualmente falham ou funcionam [Kuo e Zuo, 2003] [Maciel et al., 2010].

A lei adicional de probabilidades é a justificativa para o método SPD. Se dois ou mais eventos não tem elementos em comum, a probabilidade de que pelo menos um destes eventos irá ocorrer é a soma das probabilidades dos eventos individuais [Maciel et al., 2010]. Se dois eventos A e B tiverem elementos em comum, a união destes dois eventos, $A \cup B$, pode ser expressa como a união do evento A como evento $\bar{A}B$, onde \bar{A} representa o complemento de A . Assim, tem-se a seguinte equação para avaliação das probabilidades de $A \cup B$:

$$Pr(A \cup B) = Pr(A) + Pr(\bar{A}B) \quad (2.18)$$

Do mesmo modo, se existirem 3 eventos A , B e C , tem-se:

$$Pr(A \cup B \cup C) = Pr(A) + Pr(\bar{A}B) + Pr(\bar{A}\bar{B}C) \quad (2.19)$$

Generalizando para n eventos A_1, A_2, \dots, A_n , tem-se:

$$Pr(A_1 \cup \dots \cup A_n) = Pr(A_1) + Pr(\bar{A}_1 A_2) + \dots + Pr(\bar{A}_1 \dots \bar{A}_{n-1} A_n) \quad (2.20)$$

A Equação 2.20 expressa a probabilidade de união dos n eventos como a soma de n termos prováveis. Cada termo adicional representa a contribuição para a probabilidade de união por um evento adicional. Por exemplo, o primeiro termo, $Pr(A_1)$, é a contribuição para o primeiro evento, A_1 , na probabilidade de união. O segundo termo, $Pr(\bar{A}_1 A_2)$, representa a contribuição adicional de A_2 que não tem sido considerada por A_1 em relação a probabilidade de união, expressando a probabilidade de união de n eventos na soma das probabilidades de n termos simples. No entanto, a tarefa escolhida é como avaliar a contribuição adicional do evento adicional que não tem sido considerado por nenhum dos eventos anteriores.

A implementação da lei da adição pode ser dividida em duas partes, que podem ser chamadas de **laço interno** e **laço externo**

O laço externo do método SPD é baseado na Equação 2.20. Suponha que existam l caminhos mínimos no sistema denotados por MP_1, MP_2, \dots, MP_l . Considere que S_k denota a contribuição do k -ésimo caminho mínimo na união de todos os caminhos mínimos para $1 \leq k \leq l$. Desse modo tem-se as seguintes equações iterativas para avaliar S_k .

A expressão da contribuição do primeiro caminho será igual a probabilidade deste caminho funcionar:

$$S_1 = MP_1 \quad (2.21)$$

A expressão da contribuição do segundo caminho é igual a probabilidade do segundo caminho funcionar dado que o primeiro caminho falhou:

$$S_2 = \overline{MP_1} MP_2 \quad (2.22)$$

De modo geral, a expressão da contribuição do k -ésimo é obtida a partir da Equação 2.23.

$$S_k = \overline{MP_1} \overline{MP_2} \dots \overline{MP_{k-1}} MP_k \quad (2.23)$$

Isto possibilita que seja obtido o valor da contribuição do k -ésimo caminho por:

$$U_k = Pr(S_k), \quad 1 \leq k \leq l$$

Assim U_k é a probabilidade de que o k -ésimo caminho mínimo seja o primeiro caminho que faz o sistema funcionar. Em outras palavras, U_k é a probabilidade de que os primeiros $k-1$ caminhos mínimos sejam falhos e que o k -ésimo caminho funciona. Assim, a confiabilidade do sistema pode ser expressada como:

$$R_s = U_1 + U_2 + \dots + U_l \quad (2.24)$$

Em sucessivas iterações do laço externo, é possível encontrar mais e mais limites inferiores na confiabilidade do sistema. Quando todos os caminhos mínimos forem incluídos, a confiabilidade exata é encontrada [Kuo e Zuo, 2003].

A ordem em que os caminhos e cortes mínimos são considerados desempenha um papel importante na eficiência do algoritmo. As seguintes orientações, por ordem de prioridade, tornam o algoritmo mais eficiente [Kuo e Zuo, 2003].

1. Caminhos mínimos (ou cortes mínimos) com o menor numero de componentes devem ser considerados primeiro.

2. Entre caminhos e cortes mínimos de mesmo tamanho (por exemplo, com o mesmo número de componentes) aquele que tiver o maior número de componentes em comum com o caminho anterior deve ser considerado na sequência. Por exemplo, entre os caminhos bc , cde , adf e adg a ordenação correta para as duas primeiras orientações é $MP_1 = bc$ e $MP_2 = cde$ e os outros dois podem ser dispostos arbitrariamente.
3. Caminhos ou cortes mínimos com o mesmo número de componentes são ordenados de forma ascendente (se os componentes forem numerados) ou alfabética (se os componentes forem rotulados com letras do alfabeto).
4. Os componentes dentro de um caminho mínimo são ordenados de forma ascendente ou alfabética.

O laço interno utiliza um ou mais caminhos mínimos em cada iteração. No k -ésimo passo, quando o k -ésimo caminho mínimo for considerado, o laço interno será responsável por encontrar a expressão do evento que inclui o k -ésimo caminho mínimo, mas nenhum dos $k-1$ caminhos mínimos. Os passos a seguir ilustram a avaliação do k -ésimo termo, S_k , onde $k > 1$, usando a Equação 2.23:

1. Remove os componentes que estiverem presentes no MP_k de cada um dos caminhos mínimos anteriores. Encontra a expressão lógica de união destes $k-1$ caminhos mínimos modificados. Usa Álgebra Booleana para simplificar esta equação.
2. Inverte a expressão lógica simplificada, e então, encontra o complemento da expressão lógica simplificada. Esta expressão é a soma ou união dos produtos.
3. Expressa a expressão lógica obtida nos passos anteriores como a soma dos produtos disjuntos.
4. Multiplica cada termo da expressão lógica obtida pela expressão lógica do k -ésimo caminho mínimo. Isto gerará a expressão de S_k que representa a contribuição adicional do MP_k e é disjunto com cada um dos caminhos mínimos anteriores.

Uma vez que S_k seja obtido, U_k pode ser obtida diretamente de S_k . Quando U_k se torna disponível para $1 \leq k \leq l$, o valor de R_s pode ser encontrado facilmente.

Quando são considerados os caminhos mínimos, sucessivos valores inferiores são encontrados. Já quando são considerados os cortes mínimos, valores superiores são encontrados.

2.8 IMPORTÂNCIA DE COMPONENTES

Historicamente, Birbaum foi o precursor das discussões sobre importância de componentes em sistemas coerentes. Métricas desta natureza indicam como um componente em

particular é relevante para o sistema. Importância de componentes é um método muito útil para os projetistas, fabricantes e especialistas em manutenção para descobrir como a falha de um componente poderá afetar o funcionamento de todo o sistema e, assim, identificar os componentes fracos do sistema [Si et al., 2009].

Várias medidas estão disponíveis para estimar a importância do componente, que muitas vezes se relacionam com a contribuição de um componente para a falha do sistema. Assim, para avaliar a importância de diferentes aspectos do sistema, um conjunto de medidas tem sido definidas e largamente adotadas na prática da engenharia, como por exemplo, importância estrutural, importância para confiabilidade e a importância crítica [Kuo e Zuo, 2003] [Si et al., 2010].

Enquanto um índice pode ter por objetivo o maior ganho na melhoria da confiabilidade do sistema, outros podem oferecer a melhor assistência na concepção e otimização do sistema, ou mesmo sugerir a forma mais eficiente para operá-lo sem falha.

Para exemplificar, considere um sistema com dois componentes. Se o componente 1 for mais provável levar o sistema a falha que o componente 2, então o componente 1 é mais importante, de alguma forma, que o componente 2. Similarmente, questões mais complexas relacionadas a custo e prioridades de projeto dependem de alguma medida da contribuição de cada componente ou de um valor para a confiabilidade do sistema [Armstrong 1997].

Para melhor entendimento, considere um sistema que seja composto por componentes que falham estocasticamente e considerando ainda que os recursos disponíveis são limitados. Como a confiabilidade do sistema pode ser melhorada?

Para responder a questões como essa, o conceito de importância para componentes é apresentado. Inclusive, neste caso específico, pode ser adotado o índice denominado “importância para confiabilidade”. Esta medida provê uma lista para determinar quais componentes são mais importantes para a confiabilidade do sistema ou, em outras palavras, quais são mais críticos para a falha do sistema [Wang et al., 2004].

Considere o sistema composto por 3 componentes, apresentado na Figura 2.9. Neste caso, sem estudo de importância de componentes, apenas conhecendo os componentes individualmente, poderia ser sugerido que fossem incluídos dois componentes B redundantes, uma vez que estes componente costuma falhar. Isto, provavelmente iria melhorar o sistema, no entanto, não é garantido que fossem necessários estes dois componentes, podendo haver desperdício de recurso.

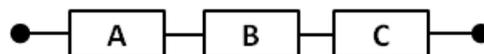


Figura 2.9 Sistema com 3 componentes em série

Para direcionar estas melhorias, e entender melhor os sistemas, várias medidas foram propostas. Algumas medidas de importância são mais úteis que outras. Na subseção seguinte será descrito o índice de importância para confiabilidade, pois este é um índice

clássico que é largamente adotado quando se trata de confiabilidade. Através de sua análise é possível identificar facilmente quais componentes têm maior impacto na confiabilidade do sistema, e que precisam ser melhorados ou aplicado redundância.

2.8.1 Importância para Confiabilidade

O índice de Importância para Confiabilidade (RI) de um componente i representa a quantidade de melhoria na confiabilidade do sistema quando a confiabilidade do componente i aumenta em uma unidade [Kuo e Zuo, 2003], podendo ser entendida como o impacto que um componente representa na confiabilidade do sistema [Si et al., 2010]. Matematicamente, a obtenção do valor da métrica RI é feita obtendo-se a derivada parcial da confiabilidade do sistema em relação a taxa de falha de cada componente individualmente [Hilber e Bertling, 2007] e pode ser representada pela seguinte equação:

$$I_i^B = \frac{\partial R_{sys}(t)}{\partial p_i(t)} \quad (2.25)$$

onde I_i^B é o índice de importância para confiabilidade (ou Importância de Birnbaum) do componente i ; p_i é a confiabilidade do componente i ; R_{sys} é a confiabilidade de todo o sistema e t é o tempo considerado para obtenção da confiabilidade.

Baseado nesta definição e observando que $0 \leq p_i \leq 1$ a importância para confiabilidade de um componente i pode ser escrita na forma:

$$I_i^R = R_s(1_i, \mathbf{p}^i) - R_s(0_i, \mathbf{p}^i) \quad (2.26)$$

onde \mathbf{p}^i representa o vetor de confiabilidade dos componentes com o i -ésimo componente removido; 0_i representa a condição quando o componente i é falho e 1_i a condição quando o componente i está sempre no modo operacional.

Pela definição anterior, a medida de RI representa um indicador para selecionar componentes que são mais interessantes para melhorar a confiabilidade do sistema. I_i^B depende somente da estrutura do sistema e da confiabilidade dos outros componentes. Em outras palavras, a RI do componente é completamente determinada pelas confiabilidades dos outros componentes, excetuando-se o componente i [Rausand e Høyland, 2004] [Kuo e Zuo, 2003].

Em uma estrutura em série, o componente menos confiável tem a maior RI. Em outras palavras, um sistema em série é tão forte quanto seu componente mais fraco. Em estruturas em paralelo, o componente mais confiável tem a maior RI. Isto significa que um sistema paralelo é tão fraco quanto o seu componente mais forte [Kuo e Zuo, 2003].

Baseado na RI, o componente com o maior índice deveria ser melhorado a fim de melhorar a confiabilidade do sistema. No entanto, esta abordagem nem sempre pode ser aplicada na prática, em alguns casos a melhoria de um componente menos confiável seria mais econômico e melhora significativamente a confiabilidade do sistema [Kuo e Zuo, 2003].

Isto é uma prática adotada para redundância, onde equipamentos com menor confiabilidade são adicionados redundantemente para aumentar a confiabilidade do sistema [Maciel et al., 2010].

Considere o exemplo apresentado na Figura 2.9, onde após analisar a importância para confiabilidade dos componentes A, B e C, foram obtidos os valores 0,4, 0,6 e 0,2 respectivamente. Baseado neste índice, seria interesse adicionar um novo componente B (redundante), como apresentado na Figura 2.10.

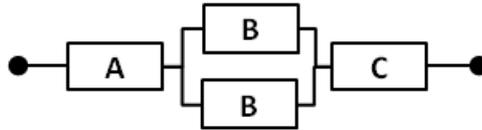


Figura 2.10 Sistema com 1 componente B redundante

Após adicionar este componente, o valor da importância para confiabilidade do componente B passa a ser 0,25, de modo que o componente A, se torna o mais importante (nesta nova estrutura).

Conforme apresentado, a importância para confiabilidade é adequada para tomar decisões de balanceamento que comumente surgem durante o projeto do sistema, uma vez que este índice indica como os componentes interagem para determinar a sua confiabilidade.

FERRAMENTA ASTRO

Este capítulo apresenta a ferramenta ASTRO, que permite suporte a análise de dependabilidade e sustentabilidade, com foco em infraestruturas de *data center*. São detalhadas as suas principais características, incluindo arquitetura, principais funcionalidades e componentes que já existiam na ferramenta. Isto é importante para conhecer a ferramenta e a estrutura que foi utilizada para implementar, adicionalmente, as métricas propostas e requeridas para realização desta pesquisa.

3.1 ARQUITETURA E FUNCIONALIDADES GERAIS

A ferramenta ASTRO foi desenvolvida para fornecer suporte à avaliação de dependabilidade e sustentabilidade, inicialmente através de modelos RBD, SPN e modelos de exergia [Marwah et al., 2010a].

Esta ferramenta foi desenvolvida sobre o núcleo da ferramenta Mercury (veja Figura 3.1), uma ferramenta que permite a análise de modelos RBD e SPN (atualmente sendo incluídos modelos de markov - MC), com várias funcionalidades implementadas. A ferramenta ASTRO utilizou todas estas funcionalidades e adicionalmente implementou modelos de alto nível (com foco em *data center*), que abstraem a utilização de RBD e SPN para os usuários não especializados, apresentando uma interface mais intuitiva para criação e análise das arquiteturas.

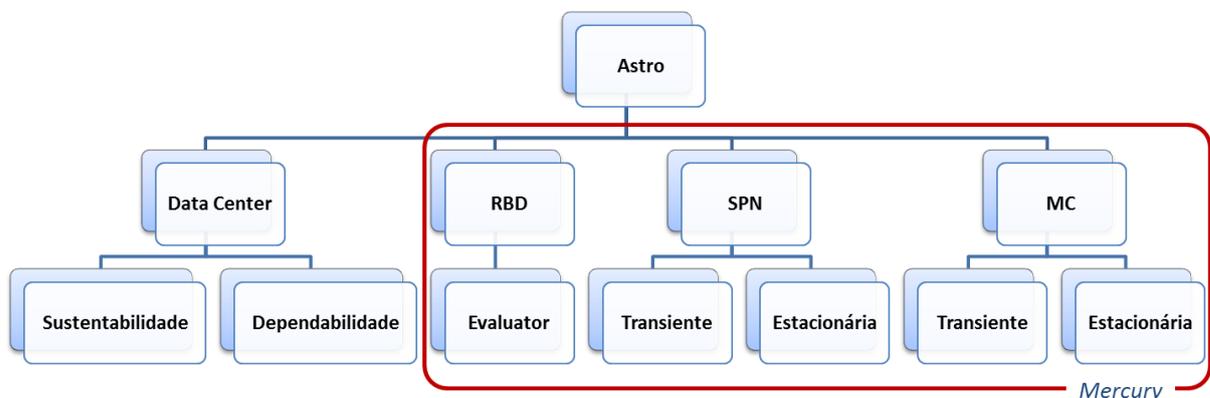


Figura 3.1 Estrutura da Ferramenta ASTRO

Além de possuir editores de alto nível para ambientes *data center*, é facultado aos usuários que conheçam os princípios de modelagem SPN e RBD, utilizar a ferramenta para avaliação de dependabilidade de sistemas genéricos, editando diretamente estes modelos, sem utilizar a visão de *data center*.

Para facilitar a utilização de múltiplos modelos é possível permutar os resultados de avaliações entre estes modelos. Isto simplifica, inclusive, os esforços na representação e avaliação de partes do sistema. Por exemplo, dado um sistema, é possível modelar uma parte mais interna (ou mais complexa) do modelo, avaliar a dependabilidade, e em seguida adicionar estes resultados como parte componente de um novo modelo (com um nível de abstração maior).

Por ter sido desenvolvida utilizando a linguagem Java, a ferramenta ASTRO é multi-plataforma, ou seja, é possível ser executada em qualquer sistema operacional que possua uma Máquina Virtual Java instalada. Exemplo desses sistemas operacionais são o Linux, Windows, Mac ou Solaris. A arquitetura da ferramenta é apresentada na Figura 3.1, sendo baseada em quatro eixos principais: (a) modelos data center, (b) modelos RBD, (c) modelos SPN e (d) modelos MC (Cadeias de Markov). Cadeias de Markov ainda está em desenvolvimento, portanto, nas seções seguintes serão detalhados os outros eixos.

3.2 EDITORES DE POTÊNCIA E RESFRIAMENTO

Os usuários não-especializados (que não tenham experiência ou conhecimento de formalismos como RBD ou SPN) têm a opção de utilizar os editores de alto nível para modelar e analisar as infraestruturas de *data center*. Nestes editores de alto nível, as infraestruturas são dispostas de forma gráfica, mostrando os componentes e a sua configuração (ou conexão). Os componentes são representados por ícones e as conexões entre os componentes é representado por uma aresta direcionada (ver a Figura 3.2). Assim, é possível representar a configuração e comunicação dos componentes dentro das estruturas.

A ferramenta ASTRO possui editores de alto nível que permitem a representação de infraestruturas de potência e resfriamento.

Além de definir a estrutura do sistema, o usuário define o modo operacional, ou seja, a maneira como o sistema funcionará. O modo operacional é determinado por um ponto de origem (*source*) e um ponto de destino (*target*), que indica o fluxo (entrada e saída) da energia ou do calor, sendo usado para obter o funcionamento do sistema.

Estes editores de *data center* (*Power* e *Cooling*) permitem que o usuário especifique a estrutura do *data center* através de ícones, que representam cada componente do sistema, descrevendo a conexão dos componentes através de arcos direcionados. Cada componente do *data center* tem informações relacionadas a dependabilidade e sustentabilidade. A Figura 3.2 mostra o editor de sistema de potência, onde é possível observar os pontos de origem e destino (*SourcePoint* e *TargetPoint*), as setas direcionadas (indicando o fluxo da energia) e os equipamentos, representados por ícones.

É importante destacar que os valores de dependabilidade, obtidos na avaliação de

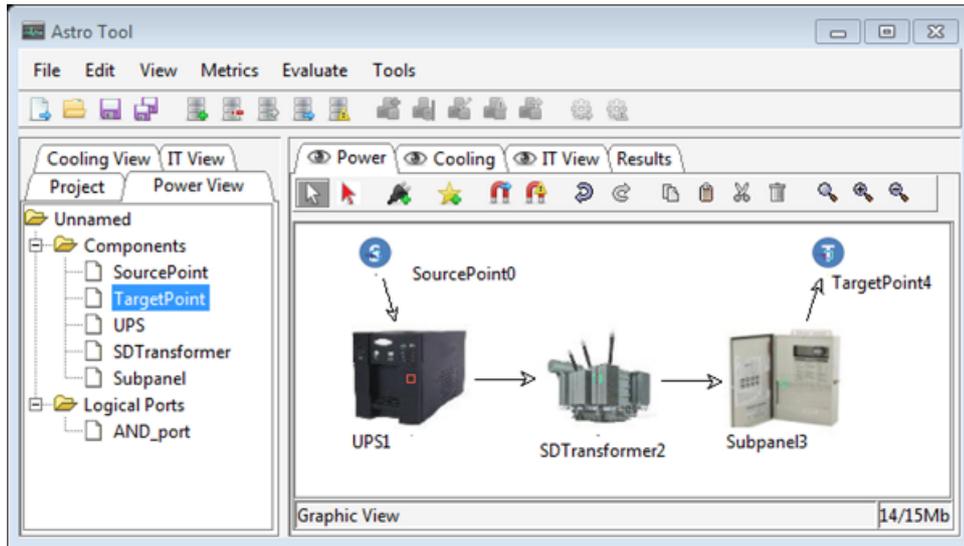


Figura 3.2 Editor de Potência na Ferramenta ASTRO

um sistema, podem ser exportados para arquivos externos (como por exemplo planilhas .xls) e também podem ser armazenados (nos modelos) para reutilização, diminuindo a complexidade das estruturas, para isto estes modelos podem ser importados para novos modelos como um simples componente.

Visão Compacta: todo o sistema é importado como um único componente (ícone). Neste caso somente um componente é adicionado no sistema e os valores das métricas anteriormente calculadas são reusados, diminuindo também a complexidade do sistema.

Visão Completa: todos os componentes que compõem o subsistema são importados, mantendo a complexidade do modelo. Esta representação provê resultados mais precisos.

A Figura 3.3 apresenta um exemplo de importação no modo compacto e no modo completo. Na figura há dois sistemas idênticos (S_1 e S_2). O sistema S_3 é formado pela importação dos sistemas S_1 (no modo compacto) e S_2 (no modo completo).

Os editores de *data center* disponibilizam duas funcionalidades principais apresentadas nas subseções seguintes. Todos os recursos presentes nos editores de RBD e SPN são acessíveis pelos editores de *data center*, possibilitando a avaliação de dependabilidade sem que seja necessário conceber, editar e avaliar diretamente os modelos RBD e SPN.

3.2.1 Conversão de modelos de data center para SPN e RBD

Os modelos RBD e SPN podem ser gerados automaticamente para as infraestruturas de *data center* modeladas no editor de alto nível (*data center*). Com isto é possível realizar

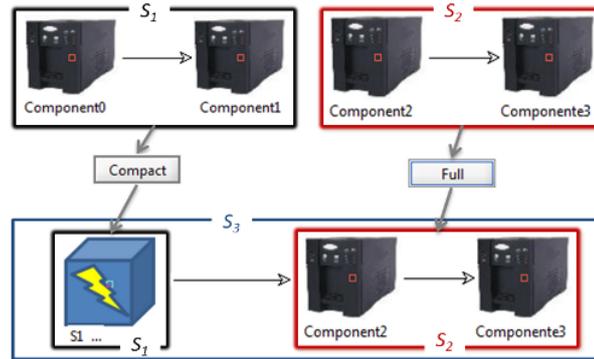


Figura 3.3 Importação de subsistemas na visão *data center*

a avaliação de dependabilidade daquelas infraestruturas.

A tradução (ver a Figura 3.4) considera o modo operacional definido pelo usuário, através dos pontos de entrada e saída comentados anteriormente.

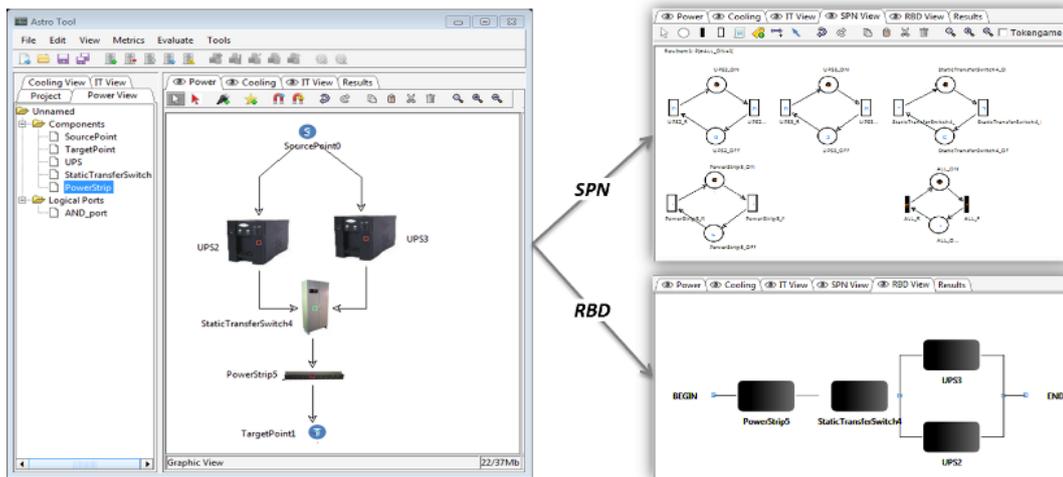


Figura 3.4 ASTRO: Conversão para RBD e SPN

O processo de tradução gera os modelos (RBD e SPN) que possibilitam a avaliação dos atributos de dependabilidade. Neste processo os componentes dispostos na visão de *data center* geram componentes RBD ou SPN e as arestas são utilizadas para definir a estrutura dos modelos.

Especificamente no procedimento de conversão para modelos RBD, cada componente gera um novo bloco, contendo os valores de tempo médio de falha e tempo médio de reparo (do componente). Estes blocos são organizados em estruturas série, paralelo ou estrutura correspondente, representando o modo operacional definido pelo usuário (veja Figura 3.4), possibilitando a avaliação de forma analítica do modelo.

No processo de conversão para modelos SPN, cada componente disposto na visão

de alto nível passa a representar um componente simples em SPN. Este componente é formado por dois lugares e duas transições, sendo que uma transição possui o tempo médio de falha associado e a outra transição possui o tempo para reparo. Os lugares representam os estados operacional e falho. Com todos os componentes criados, um novo componente simples, chamado componente agregador é adicionado, contendo, na sua transição de falha, uma “expressão de guarda”, que representa o modo operacional do sistema.

3.2.2 Avaliação do Impacto na Sustentabilidade

O impacto na sustentabilidade que um sistema de *data center* causa, pode ser avaliado através da ferramenta ASTRO. Esta avaliação adota uma abordagem baseada na avaliação do ciclo de vida (*Life-cycle assessment* - LCA [Baumann e Tillman, 2004]) para estimar o impacto na sustentabilidade de um equipamento em termos do consumo de *exergy* [Dincer e Rosen, 2007] da sua vida útil.

A abordagem LCA, comumente adotada para quantificar a sustentabilidade ambiental, basicamente divide o ciclo de vida em duas fases:

Fase *Embedded* que envolve todo o impacto relacionado às fases do projeto e da produção do equipamento (incluindo-se extração de matéria-prima, fabricação), além do impacto na cadeia de suprimento, bem como do fim-da-vida (descarte).

Fase Operacional que envolve todo o impacto relacionado a decisão durante o uso do produto ou equipamento (tais como de manutenção) e o impacto para se manter a estrutura funcionando.

Além da *exergy*, a ferramenta permite calcular o custo total de propriedade (TCO) que é composto pelo custo inicial e o custo operacional. O custo inicial corresponde aos recursos requeridos para implementar ou adquirir a infraestrutura de *data center*. O custo operacional representa o custo para manter o sistema no modo operacional. Este custo pode ser representado pela equação 3.1.

$$OC = E_{input} \times T \times E_{cost} \times A \quad (3.1)$$

onde E_{input} é a energia elétrica consumida, T é o período considerado (por exemplo: o tempo de operação), E_{cost} corresponde ao preço da energia e A corresponde à disponibilidade da arquitetura.

Para os parâmetros de sustentabilidade, a ferramenta inclui uma funcionalidade adicional, na qual é possível observar o comportamento de uma determinada métrica quando um dos parâmetros for alterado. Nesta avaliação seleciona-se o parâmetro, define-se um valor inicial, um valor final e o passo incremental e, após a execução uma faixa de valores são apresentados para métrica desejada. A Figura 3.5 mostra o resultado de uma avaliação de sustentabilidade, onde um equipamento teve a eficiência de entrega variado, considerando um intervalo especificado.

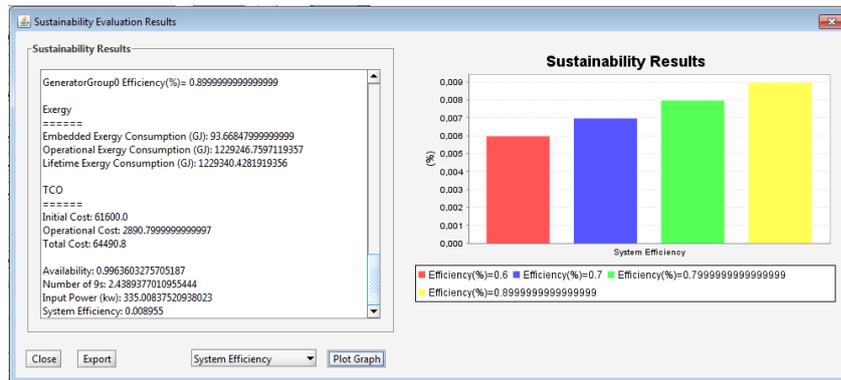


Figura 3.5 Variação de Parâmetro em Avaliação de Sustentabilidade

3.3 EDITOR E AVALIADOR SPN

O Editor e Avaliador SPN é responsável por permitir a criação, depuração e simulação de modelos SPN. A ferramenta dispõe de um editor visual, que permite a criação e análise destes modelos. Este editor deve ser utilizado por usuários que conheçam o formalismo SPN e que desejam modelar sistemas genéricos, não necessariamente de *data center*.

A avaliação de dependabilidade e desempenho no editor SPN é feita adotando técnicas de simulação, que é o processo através do qual um modelo é avaliado numericamente, e os dados resultantes desse processo são utilizados para avaliar medidas de interesse. É possível avaliar por meio de simulação transiente e estacionária [Balbo e Chiola, 1989]. Ambas as formas utilizam as mesmas funções básicas, no entanto métricas dependentes do tempo são obtidas através de simulações transientes, enquanto que as métricas de estado estacionário são resultado de simulações estacionárias.

Simulação Transiente A simulação transiente analisa o comportamento de um determinado modelo a partir do instante inicial até um determinado instante de tempo, ou seja, considera o período transiente do sistema. Este tipo de simulação pode ser utilizado para responder a perguntas do tipo: qual é a probabilidade de que após uma semana de funcionamento, o sistema ainda esteja operacional? As medidas de desempenho são computadas para o ponto final no tempo. Na Figura 3.6(a) temos um gráfico apresentando o resultado de uma simulação transiente.

Simulação Estacionária A simulação estacionária avalia o desempenho do sistema após os efeitos transitórios iniciais passarem, ou seja, em um modo de funcionamento equilibrado. A simulação do estado estacionário pode ser utilizada para responder a perguntas típicas como: qual será a largura de banda máxima de um canal de comunicação? Qual a probabilidade do sistema estar funcionando em um tempo qualquer? São questões como essas que a simulação estacionária está disposta a resolver. Na Figura 3.6(b) temos o resultado de uma simulação estacionária realizada na ferramenta ASTRO.

Ainda utilizando simulação estacionária é possível realizar uma **experimentação**, funcionalidade na qual o programa executa várias simulações automaticamente variando um determinado parâmetro da SPN. Suponha uma SPN que calcula a disponibilidade de um determinado sistema composto por vários equipamentos, é possível variar linearmente, por exemplo, o tempo de falha de um determinado equipamento e avaliar o impacto desta variação na disponibilidade do sistema final. A variação dos parâmetros se dá de forma linear, na qual são selecionados o menor valor do intervalo, o maior valor e o passo.

A Figura 3.6 apresenta os resultados de ambos os tipos de simulação, realizados na ferramenta ASTRO. A Figura 3.6(a) apresenta o resultado de uma avaliação transiente, onde é possível observar o comportamento variável durante o tempo. Já a Figura 3.6(b) apresenta o resultado de uma simulação estacionária, onde é possível observar vários valores encontrados antes de chegar ao estado estacionário.

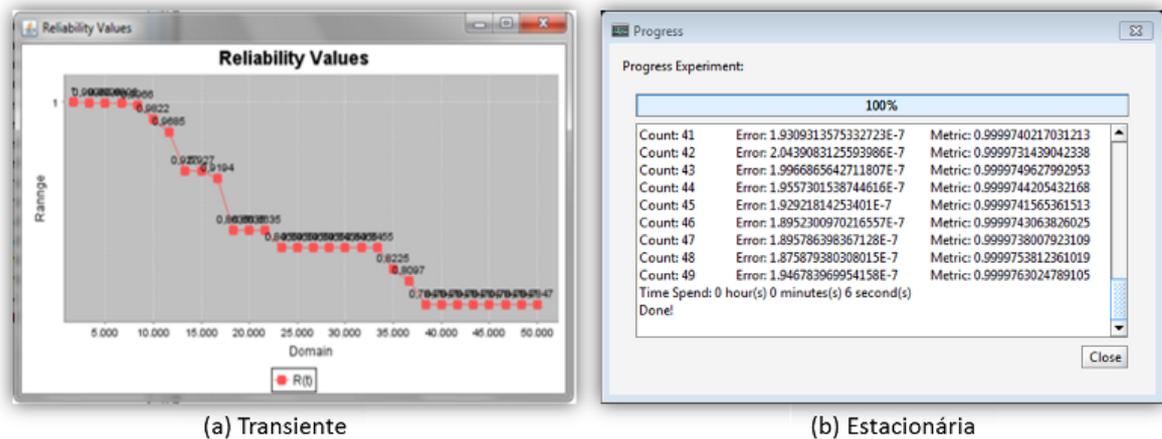


Figura 3.6 Resultados de Simulação na ASTRO

O editor para modelos SPN, em sua interface, apresenta os componentes presentes no modelo que está sendo editado: lugares, transições e métricas e a representação gráfica do modelo SPN propriamente dito, na área de modelagem. Estão dispostos ainda no editor uma barra de ícones, onde é possível acessar todas as funcionalidades disponíveis para este editor. O editor possui os componentes: lugar, transição imediata e transição temporizada.

A métrica (medidas de desempenho) definem o que deve ser computado durante a simulação. É possível calcular, por exemplo, a probabilidade de termos uma determinada quantidade de *tokens* num determinado lugar. Outros tipos de métricas também podem ser definidas, como: o número médio de *tokens* num determinado lugar ou taxa de disparo de uma determinada transição, por exemplo.

Os modelos SPN podem ser obtidos a partir de dados do editor *data center* ou criados pelo usuário diretamente. O leitor interessado deve consultar [Balbo e Chiola, 1989] e [Bause e Kritzing, 2002] para obter detalhes sobre simulação em SPN.

3.4 EDITOR E AVALIADOR RBD

O Editor e Avaliador RBD permite a modelagem e análise dos atributos de dependabilidade por meio de Diagramas de Blocos para a Confiabilidade. Adicionalmente, o ambiente RBD permite o cálculo da importância dos componentes de um determinado sistema, geração de funções estruturais e lógicas, bem como avaliação dos limites de confiabilidade do sistema. Este ambiente pode ser utilizado para modelar e analisar sistemas em geral.

A ferramenta ASTRO permite a representação deste formalismo considerando composições série e paralelo, K de N , *Bridge* e composições destas, mas também modelos simples compostos por um único componente. Na visão de RBD, o modelo contém um nó de entrada (indicando o início) e um nó de saída (indicando o fim), conforme visto na Figura 3.7.

A Figura 3.7 apresenta um modelo RBD onde é possível observar o nó inicial, representado pela palavra *BEGIN*, os componentes dispostos em composições série e paralelo, e o nó final, representado pela palavra *END*.

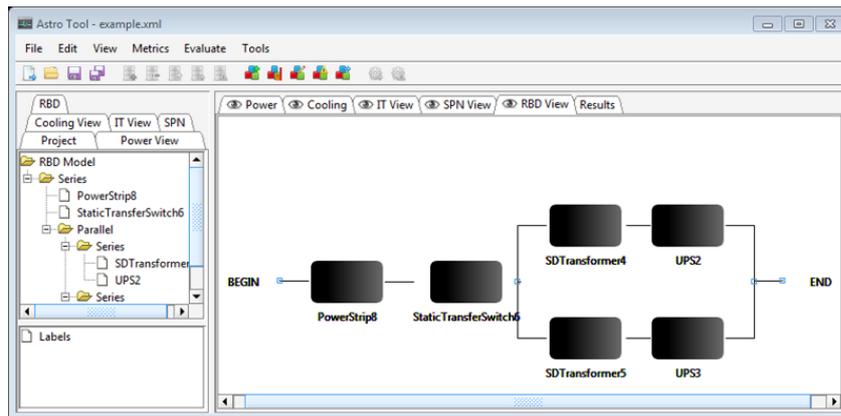


Figura 3.7 Editor de modelos RBD na ASTRO

Nos modelos RBD as métricas são avaliadas por meio de equações de forma fechadas (*closed-form*), o que permite a obtenção dos resultados de forma mais rápida, comparando-se com simulação em SPN. Os atributos disponibilidade instantânea e estacionária, confiabilidade, tempo médio para falha, tempo médio para reparo e *downtime* podem ser obtidos utilizando modelos RBD na ferramenta.

A Figura 3.8 apresenta o resultado de uma avaliação RBD. Todas as métricas disponíveis na ferramenta foram incluídas e o tempo de considerado para a confiabilidade e disponibilidade instantânea foi de 8760. Foram calculados ainda três valores intermediários (no tempo) para a confiabilidade e disponibilidade instantânea que possibilitarão a geração de um gráfico contendo a distribuição destes valores no tempo especificado.

Em virtude do tempo para se obter o valor das métricas, a utilização de RBD re-

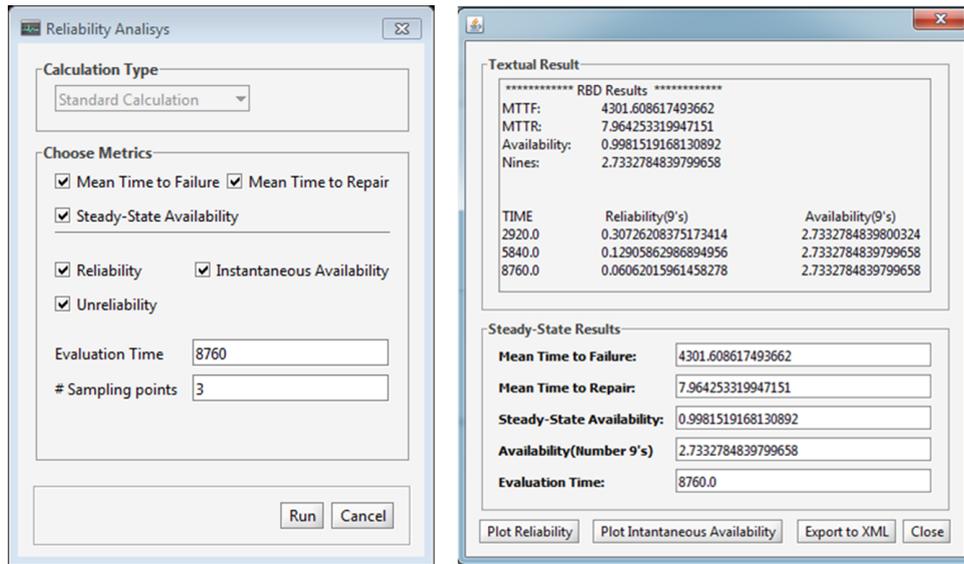


Figura 3.8 Resultado de uma avaliação em RBD

presenta uma alternativa interessante para análise de dependabilidade, no entanto há situações em que modelar estruturas utilizando RBD envolve um esforço muito maior que modelar a mesma realidade com SPN (por exemplo, dependência entre componentes). Por isso a importância de uma ferramenta que permita multi-modelos, de modo que seja possível sempre adotar o formalismo mais interessante para cada caso.

O ambiente RBD da ferramenta ASTRO possui como funcionalidades a avaliação de métricas de dependabilidade, a experimentação de cenários, a avaliação de importância dos componentes, a avaliação dos atributos de dependabilidade por aproximação, as reduções do modelo e, a geração de funções lógica e estrutural. Todas estas funcionalidades serão descritas em detalhes no Capítulo 4.

O editor e avaliador RBD, em sua interface gráfica, possui uma barra lateral esquerda, que apresenta a estrutura do modelo RBD em forma de árvore estrutural e uma barra inferior esquerda, que apresenta os rótulos (*labels*) para experimentação de cenários. A área central da ferramenta é dedicada à edição do modelo, que é disposto de forma gráfica. Além disso há também uma barra de ícones, onde é possível acessar todas as funcionalidades relacionadas a este modelo. A Figura 3.7 apresenta a interface gráfica com todos estes elementos.

Este capítulo descreve as principais funcionalidades implementadas na ferramenta para avaliação de modelos RBD e detalha cada uma destas funcionalidades, demonstrando sua aplicação.

4.1 GERAÇÃO DA FUNÇÃO ESTRUTURAL

A ferramenta ASTRO permite a geração automática da Função Estrutural, que é uma representação comportamental (sob a perspectiva de falha e funcionamento) da estrutura do sistema, na forma de função (veja Seção 2.4.1), para modelos RBD. A função estrutural de um modelo RBD é uma forma de representar matematicamente o sistema por meio da adoção de uma função algébrica.

Este método gera expressões que representam funções matemáticas discretas. Elas são adotadas para indicar um relacionamento entre os estados dos componentes e o estado do sistema. Cada expressão recebe entradas binárias que representam o estado de operação de cada componente, e retorna o estado do sistema (operacional ou falho). O estado dos componentes e do sistema são representados (na equação) pelos valores 0 (componente está falho) e 1 (componente está operacional).

Esta funcionalidade está disponível no ambiente RBD da ferramenta e pode ser utilizada para analisar o comportamento do sistema. A partir da análise da sua função estrutural, pode ser possível identificar quais componentes estão estruturados representando galgalos no sistema e que, eventualmente, poderiam ser replicados.

A Figura 4.1 apresenta um modelo RBD contendo estruturas série e paralelo. Este modelo possui quatro blocos b_1 , b_2 , b_3 e b_4 , sendo que neste modelo os blocos b_2 e b_4 formam uma estrutura em série, que está em paralelo com b_3 . Esta estrutura (contendo os blocos b_2 , b_3 e b_4) está em série com o bloco b_1 . A função estrutural gerada para este modelo é apresentada na Figura 4.2.

Observe que os blocos são representados na função conforme a estrutura do modelo, ou seja, o bloco b_1 , que está em série com o restante do modelo (conforme Figura 4.1), é representado por uma variável que é multiplicada pelo restante da estrutura.

Além de gerar a função estrutural, a ferramenta também avalia esta função. Os valores de entrada das variáveis são os estados dos blocos e o resultado é o estado do sistema (o sistema funciona ou não funciona). A Figura 4.2 apresenta o estado (operacional) do modelo da Figura 4.1 em *“The System is Working”* (O Sistema Funciona).

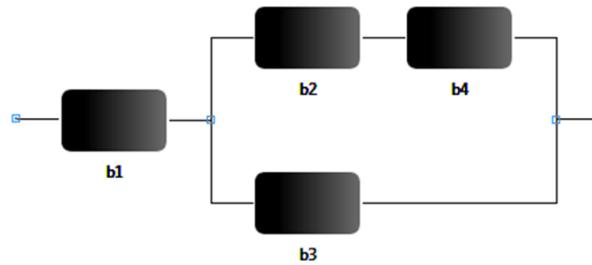


Figura 4.1 Exemplo de Modelo RBD

Utilizando a funcionalidade de avaliação da função estrutural é possível realizar testes no modelo. Para isso o usuário pode alterar o estado dos blocos (para falho, por exemplo), e re-avaliar a função estrutural. Desse modo é possível observar a relação entre a falha de um determinado componente e a falha do sistema. A interface gráfica apresenta, ainda, uma lista dos blocos marcados como falho na realização dos experimentos.

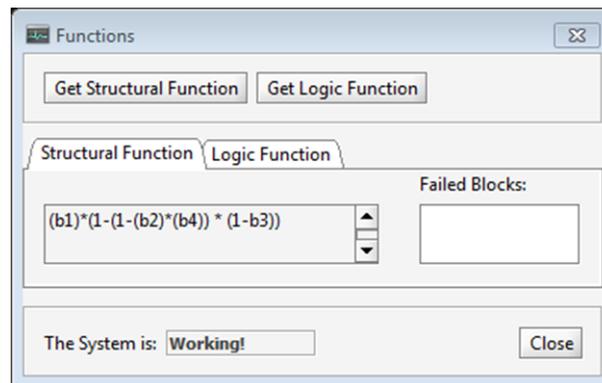


Figura 4.2 Exemplo de Geração de Função Estrutural para o RBD da Figura 4.1

A formação das expressões, no processo de geração da função estrutural, ocorre da seguinte maneira:

Componentes em série: Sejam n componentes em série e x_1, x_2, \dots, x_n as respectivas variáveis, a função estrutural ϕ desses componentes é representada por:

$$\phi = x_1 * x_2 * \dots * x_n$$

Se qualquer (algum) componente da estrutura em série estiver falho, ou seja, com o valor 0, o resultado da ϕ é 0 (zero).

Componentes em paralelo: Sejam n componentes em paralelo e x_1, x_2, \dots, x_n as respectivas variáveis, a função estrutural ϕ desses componentes é representada por:

$$\phi = 1 - (1 - x_1) * (1 - x_2) * \dots * (1 - x_n)$$

Se algum componente da estrutura estiver operacional, ou seja, estiver com valor 1 (um) o valor de ϕ será 1 (um). Somente se todos os componentes possuírem o valor 0 é que ϕ será avaliado para '0'.

4.2 GERAÇÃO DA FUNÇÃO LÓGICA

A função lógica é uma função booleana que atribui ao vetor de estados (representado através de predicados booleanos) o estado do sistema. As funções lógicas possuem o mesmo objetivo das funções estruturais: indicar o relacionamento entre os estados dos componentes e o estado do sistema (veja Seção 2.4.2), mas neste caso, utilizando operadores lógicos.

Embora as funções lógica e estrutural tenham o mesmo objetivo, há casos em que não é fácil simplificar funções estruturais [Kuo e Zuo, 2003]. Quando se tem uma função lógica a tarefa de minimização pode ser mais simples, pois nestes casos é possível aplicar técnicas de minimização booleana (álgebra booleana), para se obter, de forma mais simples, as funções lógicas minimizadas (mais fáceis de analisar).

A Figura 4.3, apresenta a função lógica correspondente ao modelo RBD apresentado na Figura 4.1. A ferramenta gerou a função lógica e em seguida avaliou o estado do sistema (operacional ou falho), baseado no estado dos componentes, conforme mostrado na Figura 4.3.

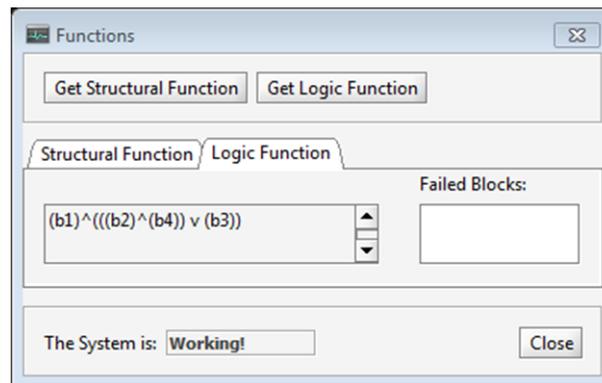


Figura 4.3 Exemplo de Geração de Função Lógica

O processo de geração da função lógica, a partir de modelos RBD, segue a seguinte abordagem:

Componentes em paralelo: As variáveis que representam o estado booleano de cada componente de um sistema em paralelo são relacionados, na função lógica, através do operador lógico \vee (OU). Sejam n componentes em paralelo, e s_1, s_2, \dots, s_n as respectivas variáveis booleanas contendo o estado correspondente ao componente, a expressão lógica que representa o funcionamento do subsistema formado por estes componentes é $s_1 \vee s_2 \vee \dots \vee s_n$.

Componentes em série: As variáveis que representam o estado booleano de cada componente de um sistema em série são relacionadas, na função lógica, através do operador lógico \wedge (E). Sejam n componentes em série e s_1, s_2, \dots, s_n as respectivas

variáveis booleanas, a expressão lógica que representa o funcionamento do subsistema formado por estes componentes é $s_1 \wedge s_2 \wedge \dots \wedge s_n$.

Assim como para a função estrutural, o gerador da função lógica também executa a avaliação desta função baseada no estado dos componentes (veja a Figura 4.3) e lista os componentes falhos.

Para auxiliar experimentos baseados nas funções (estrutural e lógica), os estados dos blocos podem ser alterados (entre operacional e falho) no editor e avaliador RBD. Isto evita a retirada de um bloco do modelo quando houver necessidade de representar a falha deste componente e assim, observar o comportamento do sistema, dada a alteração naquele componente.

A Figura 4.4 apresenta um novo exemplo de geração e avaliação da função estrutural a partir de um modelo RBD composto por 4 componentes. Observe que na Figura 4.4(a) os blocos b_2 e b_4 tiveram o seu estado alterado para falho, enquanto que os blocos b_3 e b_5 continuam no seu estado normal (operacional). O resultado da avaliação da função lógica é apresentado na Figura 4.4(b) onde é possível observar a função gerada, a lista contendo o nome dos blocos que tiveram o seu estado alterado para falho e o estado do sistema (*Failed*).

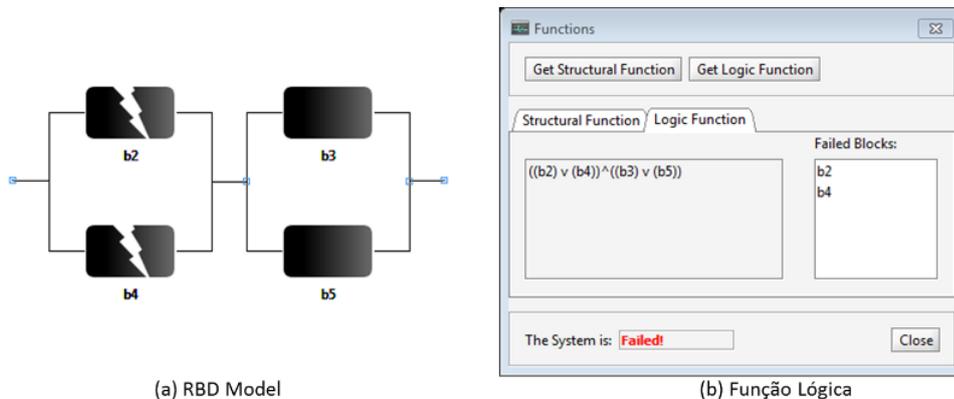


Figura 4.4 Avaliação da função lógica com bloco falhos no modelo

4.3 AVALIAÇÃO BÁSICA DAS MÉTRICAS DE DEPENDABILIDADE

As métricas de dependabilidade dos modelos RBD são avaliadas utilizando o princípio da redução do modelo, a partir de sua função estrutural. Estes métodos foram apresentadas nas Seções 2.4 e 2.7.1. Quando, em virtude da estrutura, não é possível aplicar reduções série e/ou paralelo, como é o caso da *bridge*, é aplicada a decomposição pivotal e, em seguida, reduções série e paralelo. Todas as configurações de estruturas suportadas pela ferramenta (série, paralelo, k de n (*k out of n*) e *bridge*) podem ser analisadas utilizando estas técnicas.

Para entendimento da execução do algoritmo de obtenção das métricas, considere o modelo RBD disposto na Figura 4.5(a). Este modelo é constituído por quatro blocos, onde os blocos b_2 e b_4 formam uma estrutura em paralelo, e esta estrutura está em série com os blocos b_1 e b_3 . A Figura 4.5(b) apresenta representação do modelo RBD em forma de árvore, obtida na ferramenta ASTRO.

São consideradas estrutura simples aquelas que contém apenas blocos, por exemplo, a estrutura *Parallel* da Figura 4.5(b), que contém os blocos b_2 e b_4 . Estruturas compostas são aquelas que contém estruturas (simples ou não), podendo ser combinadas com blocos, por exemplo, na Figura 4.5b onde a estrutura *Series* é considerada composta, pois além de conter blocos (b_1 e b_3), contém também outra estrutura (*Parallel*).

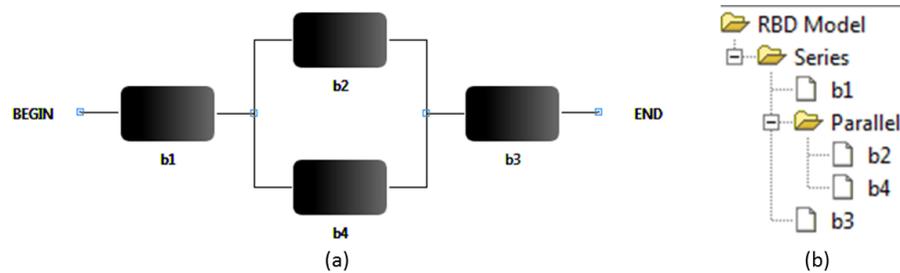


Figura 4.5 Modelo RBD constituído por blocos em série e paralelo

Na execução do algoritmo as estruturas internas simples são avaliadas primeiro e em seguida, o cálculo é aplicado para as estruturas compostas, para isso, é utilizado o resultado das estruturas simples, que passam a ser tratadas como blocos. Este procedimento é aplicado recursivamente, até se obter o valor da métrica para o sistema como um todo.

Para a avaliação do modelo RBD da Figura 4.5(a), primeiro seria calculado o valor da estrutura *Parallel*, em seguida esta estrutura seria tratada como um bloco, sendo então calculado o valor para a estrutura *Series*. Considere os blocos b_1 , b_2 , b_3 e b_4 , da Figura 4.5(a), possuindo, individualmente, a disponibilidade de 0,9 (equivalente a 90%). Para obter a disponibilidade do sistema são realizados os seguintes cálculos (considere x_1 , x_2 , x_3 e x_4 as disponibilidades dos blocos b_1 , b_2 , b_3 e b_4 , respectivamente):

1. É calculado o valor da disponibilidade para a estrutura em paralelo contendo os blocos b_2 e b_4 da seguinte forma: a disponibilidade $Disp_{b_2b_4} = 1 - (1 - x_2) * (1 - x_4) = 1 - (1 - 0,9) * (1 - 0,9) = 0,99$, ou seja, o subsistema correspondente a estrutura em paralelo possui disponibilidade de 0,99.
2. É obtido o valor para os blocos restantes que estão dispostos em série com a estrutura paralelo. A estrutura paralelo, contendo os blocos b_2 e b_4 , passa a ser tratada como um bloco único (chamado de b_p), com variável x_p contendo disponibilidade igual a 0,99. Sendo assim, a disponibilidade do modelo é dada por $Disp_{b_1b_pb_3} = x_1 * x_p * x_3 = 0,9 * 0,99 * 0,9 = 0,8019$.

Conforme avaliado, o valor da disponibilidade para o sistema representado pelo modelo da Figura 4.5 é de 0,8019 ou 80,19%.

Utilizando a avaliação básica em RBD é possível calcular as seguintes métricas de dependabilidade: tempo médio de falha do sistema, tempo médio de reparo do sistema, confiabilidade, e disponibilidade. A Figura 4.6 apresenta um exemplo de apresentação de resultados obtidos para estas métricas.

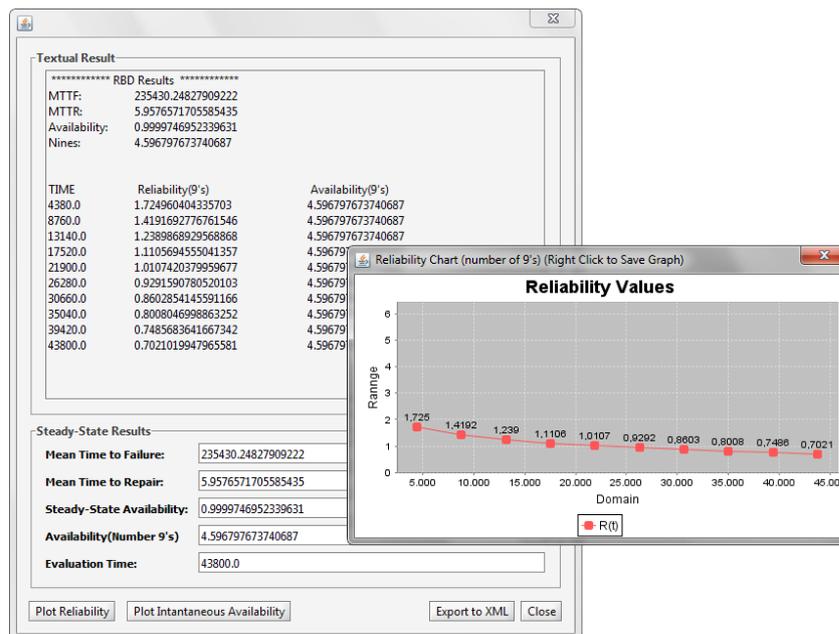


Figura 4.6 Exemplo de Avaliação Básica em RBD

Para as métricas de disponibilidade instantânea e de confiabilidade, é possível obter valores para momentos específicos de tempo entre o tempo inicial e o tempo informado para obtenção da métrica e, assim, observar o seu comportamento. Para isto, é necessário informar, no momento da avaliação, quantas amostras se deseja obter. A partir dos valores obtidos é possível gerar um gráfico onde é possível observar, os valores da métrica ao longo do tempo.

A Figura 4.6 apresenta um gráfico para a confiabilidade contendo 10 pontos. Para geração dos pontos foi informado o tempo de 43800 horas e solicitado os 10 pontos. Assim, a métrica é calculada para 10 pontos no tempo. A primeira calculada para 4380 horas, a segunda para 8760 horas e assim sucessivamente.

Além das funcionalidades já apresentadas, também é possível realizar experimentação de cenários. Na experimentação são executadas várias avaliações para a métrica de interesse, variando um determinado parâmetro. Rótulos (Figura 4.7(a)) são utilizados como variáveis que têm seu valor alterado ao longo do processo de experimentação. São selecionados qual a métrica a ser analisada, o parâmetro que será variado e informados o valor inicial, valor final e o passo para a experimentação. Com esta funcionalidade, observa-se o impacto da variação de um parâmetro no resultado da métrica calculada.

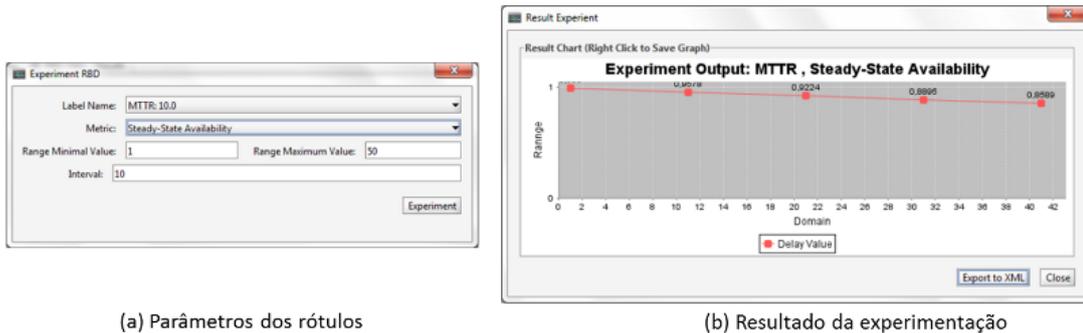


Figura 4.7 Experimentação com Rótulos em RBD

A Figura 4.7(a) apresenta a configuração dos parâmetros de uma experimentação, onde a métrica escolhida foi Disponibilidade em Estado Estacionário, e o parâmetro variado foi o MTTR de 1 hora a 50 horas com intervalo de 10 horas. A Figura 4.7(b) apresenta o resultado do experimento de forma gráfica.

4.4 AVALIAÇÃO DE DEPENDABILIDADE POR LIMITES

Os métodos para avaliação de dependabilidade em modelos RBD, no geral, fornecem valores exatos para os resultados. Para sistemas grandes e complexos, a avaliação de métricas exatas pode levar muito tempo [Kuo e Zuo, 2003]. Algumas técnicas de avaliação fornecem aproximações (ou um intervalo de valores) para a métrica. Alguns destes valores, são obtidos, normalmente, utilizando menos tempo que resolvendo, precisamente, todas as equações de forma fechada.

A Seção 2.7.2 cita alguns destes métodos e a ferramenta implementa o método da **soma dos produtos disjuntos** [Kuo e Zuo, 2003] que fornece, a cada iteração, um valor mais próximo ao valor exato para a métrica de interesse, fornecendo o valor exato na última iteração. Se adotado para caminhos e cortes, o método fornece dois valores, um superior e um inferior, que delimitam um intervalo onde está contido o valor exato da métrica.

O método soma dos produtos disjuntos é baseado na lei da adição [Kuo e Zuo, 2003] que diz que “se dois ou mais eventos não tem elementos em comum, a probabilidade de que pelo menos um desses eventos irá ocorrer é a soma das probabilidades dos eventos individuais”, ou seja, o algoritmo calcula a probabilidade de cada um dos caminhos funcionarem, dado que os caminhos anteriores não funcionaram.

4.4.1 Geração de Caminhos e Cortes Mínimos

O método soma dos produtos disjuntos utiliza os caminhos e cortes mínimos para encontrar valores aproximados para as métricas de interesse.

Dado que um caminho é um conjunto de componentes em que o funcionamento simultâneo garante o funcionamento do sistema, um caminho mínimo é um conjunto mínimo de componentes em que o funcionamento simultâneo garante o funcionamento do sistema. Um corte é um conjunto de componentes em que a falha simultânea leva o sistema a falha. Assim, um corte mínimo é um conjunto de componentes em que a falha simultânea dos componentes leva o sistema a falha [Kuo e Zuo, 2003].

Os caminhos são utilizados para obtenção dos limites inferiores, enquanto os cortes são utilizados para obtenção dos limites superiores. Assim, o número de caminhos e cortes indicam, também, quantas iterações serão necessárias para obtenção do valor exato da métrica.

A ferramenta gera os caminhos e cortes mínimos através de um método de composição recursiva. Este método gera os caminhos e cortes para estruturas simples (mais inferiores) e, em seguida, compõe os caminhos e cortes obtidos com as demais estruturas, até se obter os caminhos e cortes de todo o sistema.

Na composição dos caminhos:

- Estruturas em série com n blocos geram um único caminho mínimo contendo os n blocos;
- Estruturas em paralelo com n blocos geram n caminhos mínimos contendo 1 bloco cada caminho;
- Estruturas série ou paralelo formada por outras estruturas série e/ou paralelo, primeiro são gerados os blocos para a estruturas simples e, em seguida, os caminhos resultantes de cada estrutura simple é combinado com os caminhos resultantes das outras estruturas, compondo os caminhos deste tipo de estrutura.

Na composição dos cortes:

- Estruturas em série com n blocos geram n cortes mínimos contendo 1 bloco cada corte;
- Estruturas em paralelo com n blocos geram 1 corte mínimo contendo n blocos;
- Em estruturas série ou paralelo formadas por outras estruturas série e/ou paralelo, primeiro são gerados os blocos para a estruturas simples e, em seguida, os caminhos resultantes de cada estrutura simples é combinado com os caminhos resultantes das outras estruturas, compondo os caminhos deste tipo de estrutura.

Exemplificando, a geração de caminhos mínimos ocorre da seguinte forma:

- Para os modelos simples em série é criado um único caminho contendo todos os componentes do modelo. Por exemplo, para um modelo composto de 2 componentes, b_1 e b_2 em série, seria criado o caminho mínimo $[b_1 b_2]$.

- Para modelos paralelo simples, é criado um caminho mínimo para cada componente. Por exemplo, para um modelo composto de 2 componentes b_1 e b_2 em paralelo, seriam criados dois caminhos mínimos $[b_1]e[b_2]$.
- Para modelos compostos por estruturas série/paralelo, primeiramente é aplicada a composição para os submodelos simples e em seguida estes caminhos são compostos com os modelos mais gerais. Considere um modelo contendo o componente b_1 , que está em série com uma estrutura paralela que contém dois componentes b_2 e b_3 . Primeiramente são gerados os caminhos mínimos para a estrutura em paralelo (gerando $[b_2], [b_3]$) e em seguida, seria composto com o componente b_1 , gerando os seguintes caminhos mínimos: $[b_1b_2], [b_1b_3]$.

A Figura 4.5 apresenta um modelo RBD composto de estruturas em série e em paralelo. Para este modelo a geração dos caminhos mínimos ocorre da seguinte forma:

- São gerados os caminhos mínimos para o submodelo em paralelo composto pelo blocos b_2 e b_4 que são $[b_2]e[b_4]$.
- Em seguida, são gerados os caminhos mínimos para a estrutura em série composta pelos blocos b_1 e b_3 , gerando o caminho $[b_1b_3]$.
- Os caminhos gerados nos submodelos são multiplicados e o resultado será correspondente aos caminhos do modelo. O número total de caminhos será igual ao resultado do produto cartesiano do número de caminhos existentes ($2 \times 1 = 2$ caminhos). Na composição, o caminho formado pelos blocos em série é composto com os caminhos em paralelo, gerando os caminhos mínimos do modelo, que são: $[b_1b_3b_2], [b_1b_3b_4]$

4.4.2 Obtenção dos Limites Superiores e Inferiores

O método soma dos produtos disjuntos é um algoritmo iterativo (executado em passos) de maneira que, quanto mais passos são calculados o valor superior diminui e o valor inferior aumenta, até convergirem para o valor exato.

Na primeira iteração os valores para as métricas correspondem aos valores para o primeiro caminho (valor inferior) e para o primeiro corte (valor superior). A partir da segunda iteração, a cada novo caminho considerado, o algoritmo obtém a contribuição deste caminho para a métrica desejada. Esta contribuição, para o caminho atual, é calculada da seguinte forma:

1. São removidos dos caminhos anteriores os elementos que fazem parte do caminho atual.
2. A expressão lógica de união dos caminhos anteriores (modificados) é obtida e simplificada (usando álgebra booleana);

3. A expressão simplificada é invertida (complemento) e expressa como a soma dos produtos disjuntos;
4. Cada termo da expressão lógica é multiplicado pela expressão do caminho atual.

O valor da contribuição do caminho atual deve ser somado ao valor da métrica obtida na última iteração para obter o valor atual da métrica (mais próxima do valor exato).

De forma resumida, o método soma dos produtos disjuntos, para obtenção dos limites inferiores, calcula o valor da métrica para o primeiro caminho mínimo, obtendo-se então o primeiro valor para a métrica (inferior ao valor exato). No passo seguinte é calculada a probabilidade para o segundo caminho, gerando um novo valor para a métrica, mais próximo do valor exato. E assim sucessivamente até que sejam considerados todos os caminhos e o valor exato da métrica seja encontrado. Os valores superiores, baseados nos cortes, são obtidos de forma similar.

A Figura 4.8 apresenta o resultado de uma avaliação utilizando o método soma dos produtos disjuntos. Nesta avaliação foram obtidos 10 limites superiores e 8 limites inferiores calculados. A Figura 4.8 apresenta ainda um gráfico com os limites (superiores em vermelho e inferiores em azul), onde é possível notar que a partir do sexto passo (step 6) os valores convergem.

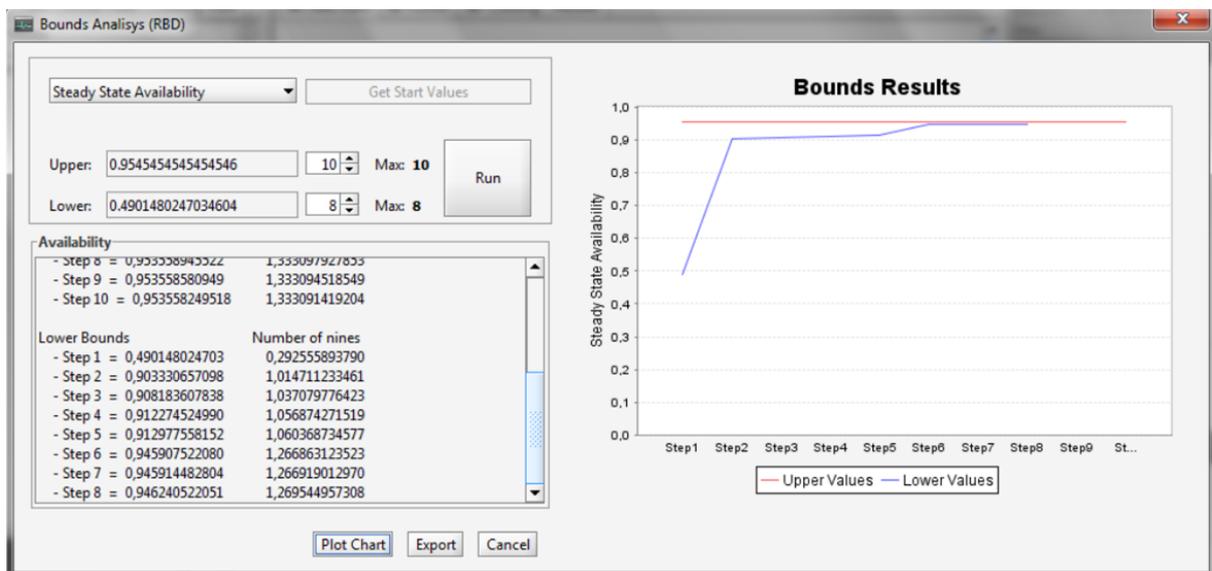


Figura 4.8 Exemplo de Avaliação por Limites/Aproximação

4.4.3 Avaliação de Modelos com Componentes Repetidos

A geração automática de modelos RBD, a partir das infraestruturas de *data center*, pode gerar modelos que contenham componentes repetidos. Isto pode acontecer por duas razões: a primeira é por especificidades do algoritmo de tradução automática de modelos

que utiliza repetição para representar algumas estruturas; a segunda são os casos em que, mesmo que concebidos manualmente, o modelo só poderá representar o sistema ou infraestrutura quando houver repetição de componentes.

A Figura 4.9(a) apresenta a estrutura de um sistema de exemplo em que, devido a sua conexão, a representação por meio de modelo RBD exige a repetição de componentes. A Figura 4.9(b) apresenta o modelo concebido contendo os blocos c_4 e c_8 repetidos. Observe que, neste caso, não há como representar o sistema através de um modelo RBD sem que sejam repetidos os componentes c_4 e c_8 ou então, repetidos os componentes c_1 e c_5 .

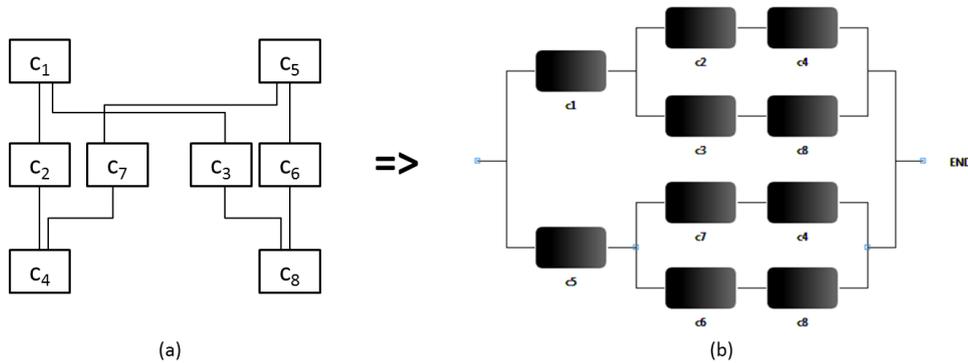


Figura 4.9 Geração de Modelo RBD com repetição de componente

Componentes repetidos aparecem em partes distintas do modelo RBD para representar conexões que são possíveis representá-las de outra forma utilizando o formalismo RBD. Modelos com este tipo de componentes exigem tratamento diferenciado para obtenção das métricas. A avaliação de modelos com componentes repetidos, utilizando a avaliação básica (Seção 4.3) gera valores inconsistentes, uma vez que considera os blocos repetidos como se fossem independentes (blocos distintos), quando na verdade, trata-se do mesmo bloco, porém representando conexões distintas.

Os modelos que contém componentes repetidos são avaliados utilizando o método soma dos produtos disjuntos (incluindo todos os caminhos) que, diferente do método básico, considera caminhos mínimos e a contribuição de cada um destes caminhos para o sistema. No cálculo da contribuição de um caminho mínimo (para a métrica) os blocos iguais encontrados nos caminhos anteriores são excluídos (veja Seção 4.4), de modo que a contribuição do bloco não é duplicada. No método, todos os caminhos são tratados como conjuntos disjuntos, evitando duplicação.

O método de obtenção dos caminhos mínimos (e cortes mínimos) em modelos com componentes repetidos pode gerar caminhos não-mínimos. Nestes casos é utilizado álgebra booleana para simplificar este conjunto de caminhos e obter somente os caminhos mínimos.

4.5 REDUÇÃO DO MODELO RBD

Modelos RBD grandes e complexos, podem ser de difícil entendimento e, por natureza, são de análise complexa. Reduzir estes modelos, possibilita diminuir a complexidade, diminuindo o tamanho, e facilita a compreensão. É possível, utilizando a ferramenta ASTRO, aplicar reduções (ou simplificações) nos modelos RBD (que possuem as taxa baseadas em uma distribuição exponencial).

Estas reduções RBD permitem obter um modelo de alto nível, a partir de um modelo detalhado, reduzindo as partes do modelo que contenham detalhamento. Com a redução é possível trabalhar com modelos mais simples, mantendo as mesmas características (do ponto de vista de dependabilidade) do modelo original.

As reduções podem ser aplicadas em estruturas série, paralelo ou *bridge*, atuando nos blocos simples que compõem a estrutura. Assim, não é possível aplicar esta redução em um bloco que forma uma estrutura série com uma estrutura paralelo. Neste caso, é possível reduzir a estrutura paralelo (para um bloco) obtendo uma estrutura em série com dois blocos simples e então aplicar a redução na estrutura série.

Na redução, o usuário seleciona qual parte do modelo deve ser reduzida. A *engine* identifica os blocos que compõe a parte do modelo (formando uma estrutura simples) e a redução é realizada. As reduções consistem em substituir a estrutura (série, paralelo ou *bridge*) por um componente único que conterà as informações dos parâmetros (taxa de falha e taxa de reparo) relativos à estrutura, agora representada por um único bloco. Reduções podem ser aplicadas sucessivamente até que o modelo se torne apenas um bloco contendo a taxa de falha (Equação 4.1) e a taxa de reparo (Equação 4.2) do sistema. Todas as equações são consideradas para distribuições do tipo exponencial.

$$\lambda_{supercomp} = \lambda_{subsistema} \quad (4.1)$$

$$\mu_{supercomp} = \mu_{subsistema} \quad (4.2)$$

onde λ é a taxa de falha e μ é a taxa de reparo.

O cálculo do tempo médio para reparo dos subsistemas utiliza a Equação 4.3, apresentada por [Dhillon 2002]. Esta equação pode ser aplicada a estruturas série, paralelo ou *bridge*.

$$MTTR = \frac{\sum_{i=1}^k \lambda_i CMT_i}{\sum_{i=1}^k \lambda_i} \quad (4.3)$$

onde k é o número de componentes, λ_i é a taxa de falha do componente i para $i = 1, 2, 3, \dots, k$ e CMT_i é o tempo requerido para manutenção corretiva ou reparo para o componente i .

A depender da estrutura e parâmetros, o cálculo do tempo médio para falha envolve várias equações. Estas equações são relacionadas em [Dhillon 2002].

O tempo médio para falha de estruturas em série são obtidos conforme a Equação 4.4.

$$MTTF = \frac{1}{\sum_{i=1}^n \lambda_i} \quad (4.4)$$

onde, λ é a taxa de falha do componente.

Já para sistemas em paralelo, segundo [Dhillon 2002], não há uma fórmula geral para cálculo do MTTF, mas sim, algumas fórmulas que podem ser adotadas, dependendo da estrutura do modelo. Das alternativas, quatro são adotadas no método da redução implementado na ferramenta:

1. Quando todos os componentes têm as mesmas taxas de falha, o valor é calculado conforme a Equação 4.5.

$$MTTF = \frac{1}{\lambda} \sum_{i=1}^n \frac{1}{i} \quad (4.5)$$

2. Quando o subsistema é composto por apenas dois componentes com taxas de falha diferentes, o valor é calculado conforme a Equação 4.6.

$$MTTF = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} + \frac{1}{\lambda_1 + \lambda_2} \quad (4.6)$$

3. Quando o subsistema é composto por apenas três componentes com taxas de falha diferentes, o valor é obtido a partir da Equação 4.7, apresentada por [Kuo e Zuo, 2003].

$$MTTF = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} + \frac{1}{\lambda_3} - \frac{1}{\lambda_1 + \lambda_2} - \frac{1}{\lambda_1 + \lambda_3} - \frac{1}{\lambda_2 + \lambda_3} + \frac{1}{\lambda_1 + \lambda_2 + \lambda_3} \quad (4.7)$$

4. Quando as taxas de reparo, além das taxas de falha, forem editadas para todos os componentes é adotada a Equação 4.8.

$$MTTF = \frac{\prod_{i=1}^n MTTF_i * n}{\sum_{i=1}^n MTR_i} \quad (4.8)$$

Em todas as equações apresentadas, λ_i representa a taxa de falha do componente i .

Em sistemas paralelos que contenham mais de três componentes, a Equação 4.7 é aplicada sucessivamente, até que se obtenha apenas dois valores, sendo um de um componente e outro o valor obtido pela equação. Para estes dois valores é aplicada a Equação 4.6 obtendo o MTTF do sistema.

Como exemplo de aplicação destas equações, suponha um sistema em série com seis componentes c_1, c_2, c_3, c_4, c_5 e c_6 que possuem taxas de falha distintas e que não tenham taxas de reparo editadas. Para obtenção do $MTTF$ desse sistema, inicialmente é aplicada a Equação 4.7 para os componentes c_1, c_2 e c_3 , obtendo o $MTTF_{123}$. Em seguida é aplicado a Equação 4.7 considerando o valor obtido em forma de taxa ($\frac{1}{MTTF_{123}}$) e os

componentes c_4 e c_5 obtendo o $MTTF_{12345}$. Por ultimo é utilizada a Equação 4.6 para $1/MTTF_{12345}$ e c_6 , obtendo o $MTTF$ do sistema em paralelo ($MTTF_{123456}$).

A Figura 4.10 apresenta a aplicação de redução de um modelo RBD que é composto por quatro componentes (b_1, b_2, b_3 e b_4), onde os blocos b_1 e b_2 estão em série com uma estrutura paralelo constituída dos blocos b_3 e b_4 . Uma redução em série (a) foi inicialmente aplicada aos blocos b_1 e b_2 , gerando o bloco *subModelSeries9* que contém as informações de taxa de falha e de reparo da estrutura formada pelos componentes b_1 e b_2 . Em seguida foi aplicada redução em paralelo (b) nos blocos b_3 e b_4 , gerando o bloco *subModelParallel10* (com taxas correspondentes a estrutura em paralelo); Após foi aplicada novamente redução em série (c), agora nos blocos *subModelSeries9* e *subModelParallel10* obtendo um modelo constituído de apenas um bloco (d) (*subModelSeries11*). Este bloco contém as informações de falha e reparo correspondentes ao modelo original, no entanto é mais simples (possui uma abstração maior).

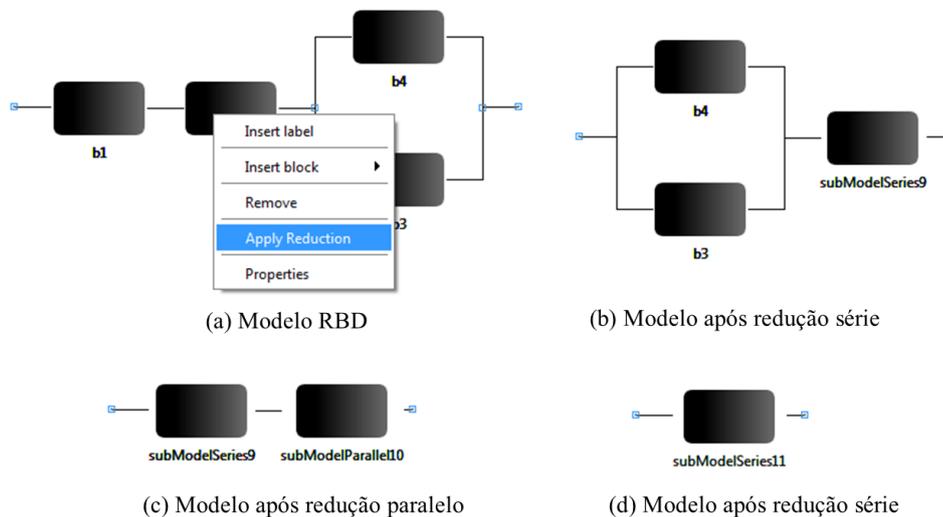


Figura 4.10 Exemplo de Redução de Modelo RBD

Considere que na Figura 4.10(c) os dois blocos *subModelSeries9* e *subModelParallel10*, possuem $MTTF = 1000$ horas e $MTTR = 4$ horas. Após aplicar a redução, ou seja, na Figura 4.10(d) os valores para $MTTF$ e $MTTR$ do bloco único serão, respectivamente 250.000 horas e 4 horas.

4.6 AVALIAÇÃO DE IMPORTÂNCIA DE COMPONENTES

A importância de componentes pode ser avaliada em termos de importância para a confiabilidade, de importância para a disponibilidade e destas relacionadas com custos.

As subseções seguintes apresentam o que representa cada um destes índices, em termos de medida e quando podem ser utilizados.

4.6.1 Importância para a Confiabilidade

Importância para a Confiabilidade (RI ou Birnbaum's Importance) do componente i é igual a melhoria da confiabilidade do sistema, quando a confiabilidade do componente é melhorada [Birnbaum 1969] [Kuo e Zuo, 2003]. A importância de confiabilidade ($I_{(i/t)}^B$) de um componente i de um sistema s é dada por:

$$I_{(i/t)}^B = \frac{\partial R_{sys}(t)}{\partial p_i(t)} \quad (4.9)$$

onde $I_{(i/t)}^B$ é o índice de importância para a confiabilidade (ou importância de Birnbaum) do componente i no tempo t ; p_i é a confiabilidade do componente i e R_{sys} é a confiabilidade do sistema.

O avaliador de importância dos componentes executa os cálculos mostrados na Seção 2.8.1 de forma simplificada, ao invés de calcular a derivada parcial mostrada na definição padrão, a ferramenta executa o cálculo mostrado na Equação 4.10.

$$I_i^R = R_s(1_i, \mathbf{p}^i) - R_s(0_i, \mathbf{p}^i) \quad (4.10)$$

onde \mathbf{p}^i representa o vetor de confiabilidade dos componentes com o i ésimo componente removido; 0_i representa a condição quando o componente i é falho e 1_i a condição quando o componente i está funcionando.

Em outras palavras, a importância para a confiabilidade de cada componente é obtida da seguinte forma: é calculado o valor para a confiabilidade do sistema quando o componente está funcionando e em seguida é calculado este valor quando o componente está falho; o valor da importância é dado pela diferença entre o valor quando o componente funciona e quando ele falha.

A Figura 4.11(a) apresenta um sistema composto por seis componentes distribuídos em estruturas série e paralelo. Cada um dos componentes é avaliado considerando sua importância para a confiabilidade. A Figura 4.11(b) apresenta o resultado da avaliação da importância para a confiabilidade de cada componente do sistema. O parâmetro de entrada para a avaliação é o tempo, enquanto o resultado, contendo o valor da importância de cada componente é apresentado tanto de forma textual como de forma gráfica. Os resultados são apresentados como valores absolutos e normalizados em relação ao de maior valor, utilizando a Equação 4.11:

$$In_i = \frac{I_i}{I_x} \quad (4.11)$$

onde In_i é o índice normalizado para o componente i ; I_i é o valor do índice não normalizado para o componente i e, I_x é o valor do maior índice não normalizado entre os componentes.

A Tabela 4.1 apresenta o valor da confiabilidade dos componentes do modelo apresentado Figura 4.11. São listados os valores da confiabilidade do sistema com o componente

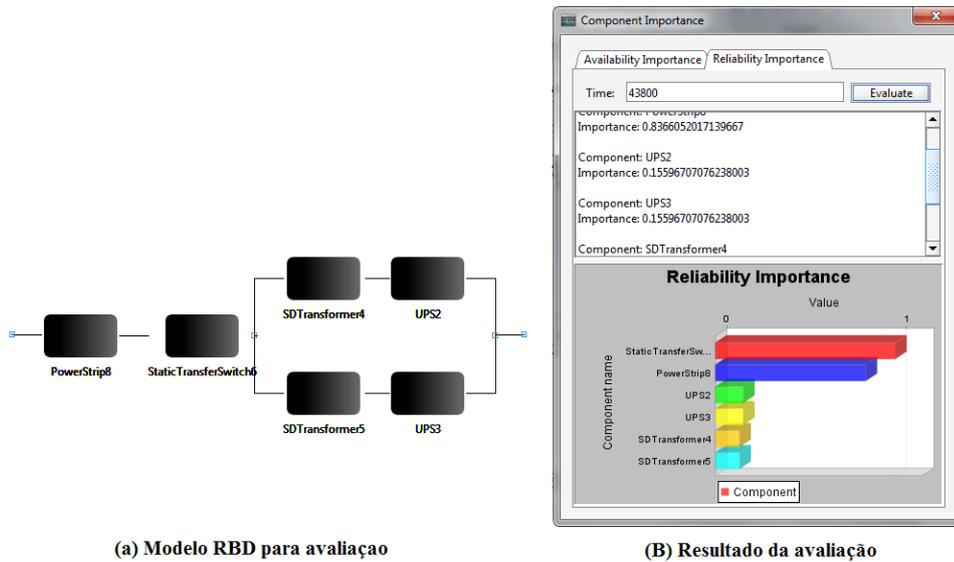


Figura 4.11 Análise de importância para a confiabilidade

Tabela 4.1 Resultado da análise de importância para confiabilidade

Componente	$R_s(1_i, \mathbf{p}^i)$	$R_s(0_i, \mathbf{p}^i)$	I_i^R
PowerStrip8	0,962641	0,000000	0,962641
StaticTransferSwitch6	0,997608	0,000000	0,997608
UPS2	0,963240	0,924555	0,038685
UPS3	0,963240	0,924555	0,038685
SDTransformer	0,962141	0,924555	0,037585
SDTransformer	0,962141	0,924555	0,037585

i operacional, com o componente i falho e, o valor da importância para a confiabilidade do componente i . O tempo de avaliação é 8760 horas; $R_s(1_i, \mathbf{p}^i)$ é o valor quando o componente está operacional; $R_s(0_i, \mathbf{p}^i)$ é o valor da confiabilidade quando o componente está falho; I_i^R é o valor para o índice.

4.6.2 Importância para a Disponibilidade

A importância para a confiabilidade considera apenas o tempo médio para falha dos componentes, pois é uma métrica voltada para confiabilidade. Com a ferramenta ASTRO é possível realizar a análise do índice de importância para a disponibilidade (I_i^A). Este índice é baseado no índice de importância para a confiabilidade e destaca quais componentes tem um maior impacto na disponibilidade do sistema, ou quais componente que, aumentando sua disponibilidade, aumentarão mais significativamente a disponibilidade do sistema.

O avaliador de importância para a disponibilidade executa cálculos baseados no cálculo

de importância para a confiabilidade. No entanto, ao invés de utilizar a confiabilidade, a métrica utilizada é a disponibilidade, incluindo, além do tempo médio para falha, o tempo de recuperação dos componentes. O cálculo executado pela ferramenta é mostrado na Equação 4.12, e os resultados apresentados são normalizados pelo maior valor, conforme Equação 4.13.

$$I_i^A = A_s(1_i, \mathbf{p}^i) - A_s(0_i, \mathbf{p}^i) \quad (4.12)$$

onde \mathbf{p}^i representa o vetor de disponibilidades dos componentes com o i ésimo componente removido; 0_i representa a condição quando o componente i é falho e 1_i a condição quando o componente i está no modo operacional.

$$In_i = \frac{I_i}{I_x} \quad (4.13)$$

onde In_i é o índice normalizado para o componente i ; I_i é o valor do índice não normalizado para o componente i e, I_x é o valor do maior índice não normalizado entre os componentes.

No cálculo deste índice não existe o parâmetro de entrada tempo, uma vez que é calculada a disponibilidade em estado estacionário.

Os resultados são apresentados de forma similar àqueles do cálculo para a importância para a confiabilidade, e são apresentados na Figura 4.12, que apresenta também os valores normalizados da importância para a disponibilidade. A relação entre os componentes mais importantes também é mostrada através de um gráfico.

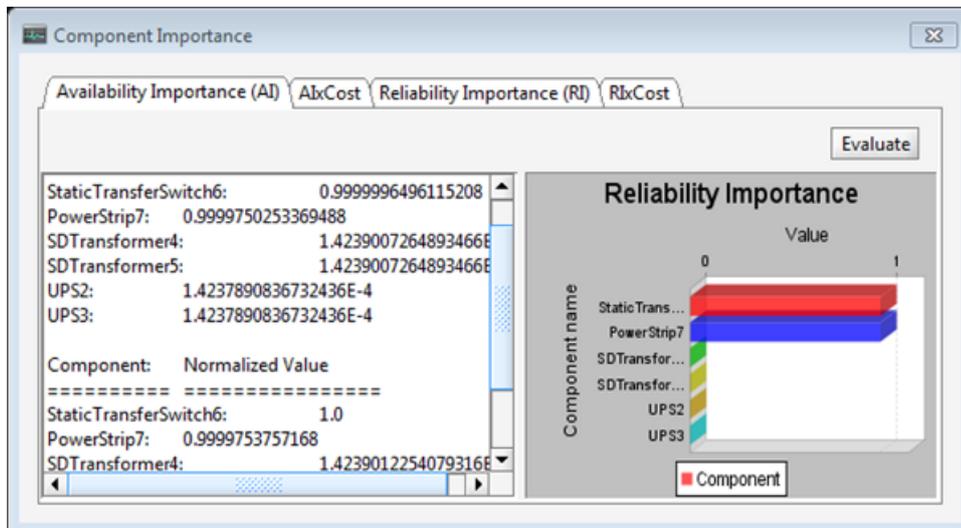


Figura 4.12 Resultado da análise de importância para a disponibilidade

Na Tabela 4.2 estão os componentes do modelo apresentado Figura 4.11 e os valores para a importância para a disponibilidade dos respectivos componentes.

Tabela 4.2 Resultado da análise de importância para disponibilidade

Componente	I_i^A
StaticTransferSwitch6	0,9999996496115208
PowerStrip7	0,9999750253369488
SDTransformer4	1,4239007264893E-4
SDTransformer5	1,4239007264893E-4
UPS2	1,4237890836732E-4
UPS3	1,4237890836732E-4

4.6.3 R_{Ic} - Importância para a Confiabilidade e Custos

Os índices até agora considerados apresentam alternativas interessantes para se quantificar o impacto de um componente no sistema, geralmente focando em melhoria ou replicação de componentes. Substituir um componente ou mesmo replicar envolve custos, que são desconsiderados nos índices apresentados até então, pois focam apenas nas métricas de dependabilidade. O impacto nos custos, causados pelo processo de melhoria de um sistema, é um fator relevante e algum índice de importância poderia considerá-lo.

O índice de RI, mesmo sendo um bom indicador para melhorar a confiabilidade, relaciona componentes candidatos a melhorias independentes de custos, ou seja, considera que os recursos são ilimitados, o que na realidade poucas vezes é verdade. Investir grandes quantias, para melhorar um componente indicado como o mais importante pode não ser viável.

Para criar uma relação entre custos e confiabilidade, foi definido um índice de *importância para a confiabilidade e custos* (R_{Ic}). Este índice visa quantificar a importância do componente para a confiabilidade do sistema relacionando o custo de aquisição do equipamento.

O método proposto provê uma lista ordenada de componentes indicando qual causará mais impacto (na confiabilidade do sistema) quando melhorado (normalmente replicado) considerando o custo do componente em relação aos outros. Adicionalmente ao índice de importância para a confiabilidade, este índice tenta quantificar o aumento que a melhoria de um componente impactará na melhoria da confiabilidade do sistema, investindo menos recursos que em outras soluções. A Equação 4.14 demonstra esta relação:

$$RIC_i = I_i^B * \left(1 - \frac{C_i}{C_{sys}}\right) \quad (4.14)$$

onde RIC_i é o índice de importância proposto; I_i^B é o valor (não normalizado) da importância para a confiabilidade do sistema i ; C_i é o custo de aquisição do componente i ; e C_{sys} é o custo de aquisição do sistema.

A Figura 4.13 apresenta um modelo RBD onde o valor do índice proposto para o *componente 1* é calculado conforme apresentado na Equação 4.15:

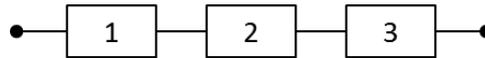


Figura 4.13 RBD com 3 componentes em série

$$RIc_1 = (1_1 * p_2 * p_3 - 0_1 * p_2 * p_3) * (1 - \frac{C_1}{C_{sys}}) \quad (4.15)$$

onde p_i é o valor da confiabilidade do componente i .

RIc_i depende da estrutura do sistema, da confiabilidade dos outros componentes, do custo do componente de interesse e do custo do sistema como um todo. Ele é independente do valor da confiabilidade $p_i(t)$ do componente i .

Em uma estrutura em série, considerando importância para a confiabilidade, o componente menos confiável é o que tem o mais alto valor de RI . Já quando é considerado RIc , o componente mais importante pode ser um outro que tenha um valor alto para RI (mas não o mais alto) e um custo pequeno (quando comparado aos outros), como pode ser observado na Tabela 4.3.

Em suma, o componente mais importante é aquele que tem alto RI combinado com baixo custo, quando comparado com os outros componentes. A Tabela 4.3 apresenta, para os três componentes, os valores da métrica de confiabilidade (tempo igual a 8760 horas), o custo de aquisição, o índices de importância para a confiabilidade e o índices de importância para a confiabilidade e custos. Considerando os resultados apresentados na Tabela 4.3, o componente mais importante (confiabilidade e custo) é o *Componente 3*. É possível observar que o resultado é diferente daquele indicado pela importância para a confiabilidade (*Componente 2*).

Tabela 4.3 Valores dos índices calculados para o modelo da Figura 4.13

	Component 1	Component 2	Component 3
Confiabilidade (8760 horas)	0,80	0,60	0,65
Custo de Aquisição	95,00	50,00	35,00
RI	0,39	0,52	0,48
RIc	0,18	0,37	0,39

A Figura 4.14 apresenta um modelo RBD, composto por 4 componentes, dispostos em estruturas série e paralelo. Na avaliação do índice de RI , considerando o período de 7650 horas (1 ano), o componente *StaticTransferSwitch* aparece como o mais importante, independente do custo dos equipamentos. Estes valores podem ser visualizados na Tabela 4.4.

Os valores apresentados para RI na Tabela 4.4 não variam se o valor de qualquer equipamento mudar, ou seja, se o equipamento *StaticTransferSwitch*, que tem o maior

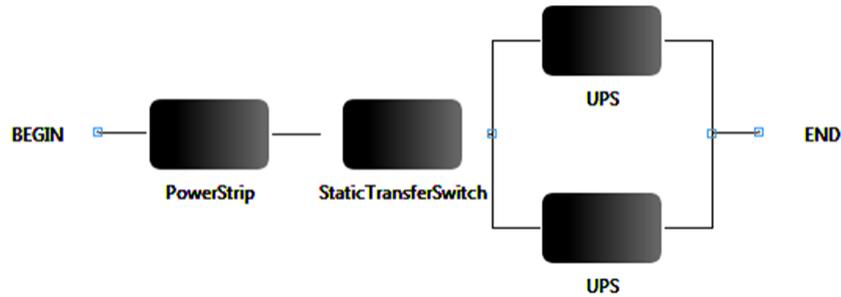


Figura 4.14 Modelo RBD com 4 componentes série e paralelo

Tabela 4.4 Valores do índice *RI* calculados para o modelo da Figura 4.14

	PowerStrip	StaticTransferSwitch	UPS	UPS
Custo de Aquisição	200,00	800,00	3600,00	3600,00
RI	0,963	0,998	0,0331	0,0331

valor de *RI*, fosse 5000,00, mesmo assim, este seria o indicado como gargalo (conforme este índice).

Na Figura 4.15 é apresentado a variação do índice de *RIc* conforme o valor do equipamento *PowerStrip* aumenta. É possível perceber que, quando este equipamento custa 200, aparece como o mais indicado para melhoria (ou redundância, baseado no índice *RIc*), no entanto, quando o seu valor passa de 500, já não aparece como o mais indicado a melhoria, pois já passa a ter um custo elevado, considerando o valor e confiabilidade dos outros equipamentos.

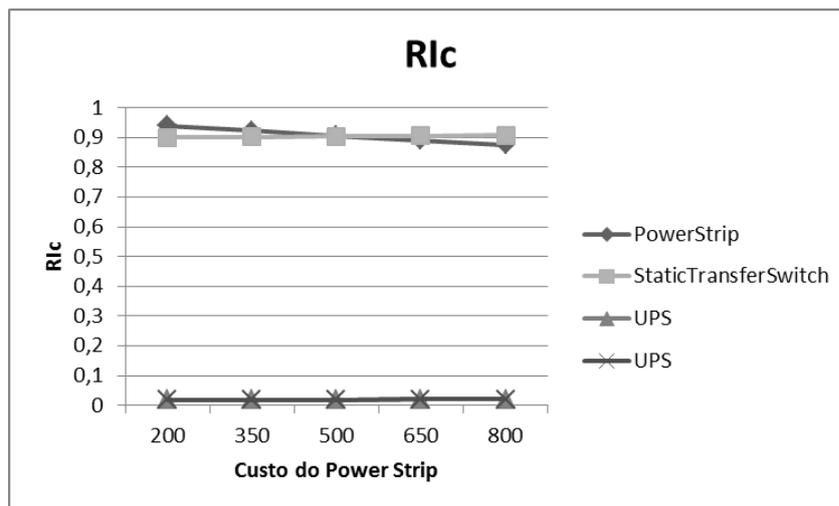


Figura 4.15 Variação da métrica *RIc* conforme variação de custo

Este índice tem grande importância pois, considerando apenas importância para a

confiabilidade, aplicar redundância no componente mais importante pode ser muito caro, comparando-se com outros componentes. Neste caso, a melhoria de um componente menos importante (no contexto de RI) é mais econômico e garante uma melhoria significativa na confiabilidade do sistema.

Como apresentado, este índice se mostra uma opção interessante para auxiliar projetistas a tomarem decisões na fase de projeto, pois indica como os componentes contribuem para a confiabilidade do sistema, além de indicar onde aplicar melhorias.

4.6.4 AIC - Importância para a Disponibilidade e Custos

Este índice corresponde ao índice de importância para a disponibilidade, combinado com a relação custo de aquisição (do mesmo modo que o índice de importância para a confiabilidade e custos). Ou seja, este índice considera os impactos no custo de aquisição dos equipamentos ao se calcular o índice de importância para a disponibilidade do componente.

Este índice considera o custo de aquisição e a disponibilidade em estado estacionário, de modo que o tempo não entra como parâmetro. Mesmo sendo um índice que avalia importância para a disponibilidade e custos, o índice não quantifica o impacto dos custos de manutenção.

O índice é calculado conforme a Equação 4.16.

$$AIC_i = I_i^A * \left(1 - \frac{C_i}{C_{sys}}\right) \quad (4.16)$$

onde AIC_i é o índice de importância proposto; I_i^A é o valor (não normalizado) da importância para a disponibilidade do sistema i ; C_i é o custo de aquisição do componente i e C_{sys} é o custo de aquisição do sistema.

AIC_i representa um bom indicador para selecionar quais componentes são melhores candidatos para melhorar a disponibilidade do sistema investindo uma quantia menor.

4.7 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas os principais métodos e métricas adotados, propostos e implementados na ferramenta.

A geração das funções lógica e estrutural podem ser adotadas para indicar o relacionamento entre o estado dos componentes e o estado do sistema. A função lógica pode ser adotada ainda, quando não for simples minimizar a função estrutural, para se obter a função minimizada através de álgebra booleana. Estas funções podem ser avaliadas para se obter o estado do sistema (operacional ou falho) baseado no estado dos componentes.

A avaliação básica das métricas de dependabilidade por meio de RBD provém meios de avaliar a dependabilidade de sistemas (e infraestruturas) considerando *closed-forms*,

obtendo o resultado bem mais rápido, se comparado com simulação.

A avaliação de dependabilidade por limites permitiu a expansão do ambiente RBD, possibilitando a avaliação de uma gama maior de modelos. Com a utilização deste método, modelos grandes e complexos podem ser avaliados, obtendo valores aproximados (ou um intervalo delimitado por um valor superior e um valor inferior) em menos tempo que obter o valor exato da métrica. Adicionalmente, o método adotado na avaliação por limites possibilitou a análise de modelos com componentes repetidos.

Reduções do modelo RBD possibilitam a obtenção de um modelo de alto nível, a partir de um modelo que represente o sistema detalhadamente. Com a redução é possível trabalhar com modelos mais simples, mantendo as mesmas características (do ponto de vista de dependabilidade) do modelo original.

As métricas de importância de componentes permitem a avaliação de sistemas sob o aspecto do impacto de cada componente para a disponibilidade e para a confiabilidade, automatizado na ferramenta. Os índices propostos considerando custos representam uma alternativa interessante, uma vez que leva em consideração questões financeiras, nem sempre abordadas em análise de importância de componentes.

A definição e implementação destes métodos e métricas, ampliam as possibilidades de avaliação de dependabilidade e permitem a automatização da avaliação das métricas aqui abordadas.

CAPÍTULO 5

ESTUDO DE CASO

Neste capítulo inicialmente será apresentada a metodologia proposta e utilizada para análise de importância de componentes considerando dependabilidade e custo, descrevendo os passos que devem ser executados. Em seguida, serão apresentados os estudos de caso envolvendo análise de importância e avaliação de dependabilidade por limites realizados como resultado da pesquisa.

O primeiro estudo de caso utiliza o índice de importância da confiabilidade, sendo analisadas sete infraestruturas de potência de *data center*. No segundo estudo de caso é utilizado o índice de importância para confiabilidade e custos, a fim de demonstrar a aplicabilidade da métrica, provendo mecanismos que permitam ao projetista melhorar o projeto das infraestruturas. Após, é apresentado o terceiro estudo de caso, onde o método de análise por aproximação (limites), mostrando que a utilização deste método é prática e pode prover resultados satisfatórios sem que se obtenha a métrica exata.

A metodologia para análise de importância de componentes considerando custos é adotada nos Estudos de Caso I e II, que tratam da aplicação destes índices. O mesmo não ocorre para o terceiro estudo de caso, onde o foco é em análise de dependabilidade por limites.

Em todos os estudos de caso, para modelagem (utilizando modelos RBD) e análise das infraestruturas, foi adotada a ferramenta ASTRO [Silva et al., 2010]. Esta ferramenta permite modelagem RBD, incluindo análise de dependabilidade e desempenho. É importante destacar que os valores para *MTTF* e *MTTR* dos equipamentos básicos foram obtidos de [Marwah et al., 2010a] e [IEEEGoldBook 1997].

5.1 METODOLOGIA PARA AVALIAÇÃO DE IMPORTÂNCIA DE COMPONENTES CONSIDERANDO CUSTOS

A Figura 5.1 apresenta uma visão geral da metodologia definida para análise de importância de componentes considerando custos. Esta metodologia foi adotada para análise em infraestruturas de potência de *data center*. Os passos apresentados representam as atividades que devem ser realizadas em cada etapa da análise. Particularmente, neste trabalho, o processo é conduzido adotando Diagramas de Bloco para a Confiabilidade, para avaliação da dependabilidade e importância de componentes do sistema. Esta metodologia é genérica e pode ser adotada por outros modelos, como Redes de Petri.

5.1 METODOLOGIA PARA AVALIAÇÃO DE IMPORTÂNCIA DE COMPONENTES CONSIDERANDO CUSTOS

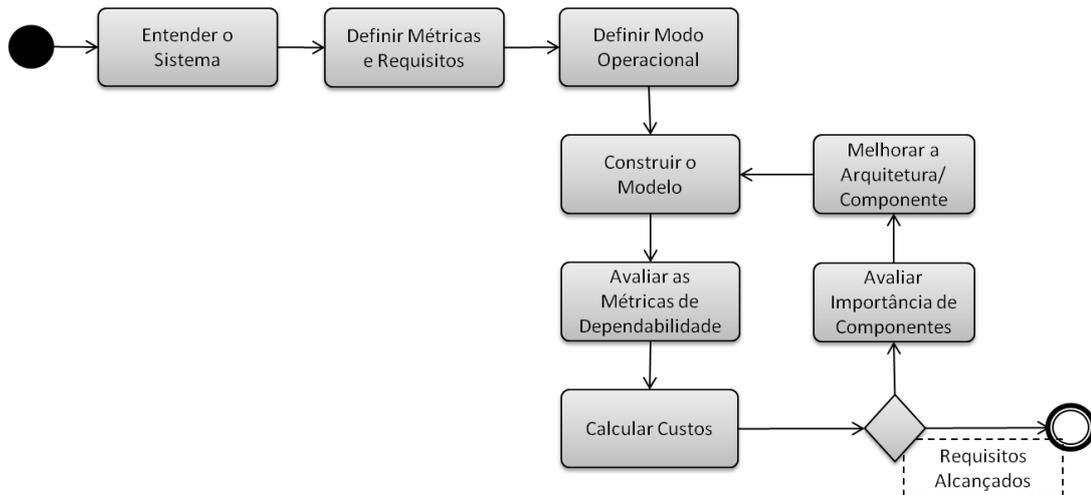


Figura 5.1 Metodologia para Avaliação de Importância de Componentes

O primeiro passo da metodologia visa entender o sistema, seus componentes, sua estrutura, suas interfaces e interações. Como resultado deste passo, é gerado um documento contendo a descrição detalhada do sistema a ser analisado.

No passo seguinte deve ser definido o conjunto de métricas a serem avaliadas (por exemplo disponibilidade, confiabilidade) e os requisitos para aceitação da estrutura baseada no valor das métricas (por exemplo, disponibilidade de 99%) e nos recursos disponíveis para investimento.

O terceiro passo visa definir o modo operacional do sistema, que estabelece o modo como o sistema é considerado funcionando (fornecendo o serviço para o qual foi projetado), ou seja, que componentes precisam estar funcionando para que o sistema esteja funcionando. Para esta atividade, o projetista deve adotar um diagrama de blocos estruturais, para melhor visualizar os componentes do sistema e as suas relações, representando tanto o seu comportamento correto quanto o seu modo de falha [Kuo e Zuo, 2003].

O modo operacional é requerido para a construção do modelo (quarto passo da metodologia), que consiste em modelar o sistema, utilizando o formalismo escolhido (por exemplo RBD ou SPN), de modo a representar o sistema. No caso de sistemas que possuem mais de um modo operacional, é necessária a criação de um modelo para cada modo operacional.

Uma vez que os modelos sejam criados, as métricas de dependabilidade devem ser avaliadas (quinto passo). No passo seguinte é realizado o cálculo dos custos (o projetista poderá optar em calcular o custo para aquisição - TCA ou o custo total de propriedade - TCO) relacionados à infraestrutura. Se o valor estimado para as métricas (dependabilidade e custos) estiverem dentro dos requisitos, a infraestrutura é aceita e o processo finalizado.

Caso os requisitos não sejam atendidos, a análise de importância de componentes é realizada (sétimo passo).

Neste passo, são identificados aqueles componentes que representam gargalos (de algum modo, conforme importância analisada) do sistema. Como produto, este passo provê uma lista de componentes indicando aqueles que podem ser melhorados. Estes componentes podem ser melhorados pela substituição do componente por um componente melhor (em algum aspecto de interesse), pela reorganização da estrutura, bem como pela adição de um componente redundante.

Após o projetista selecionar a forma de melhorar o sistema (por exemplo, aplicando redundância), retorna ao passo de construção do modelo (quarto passo), onde é remodelado o sistema, agora considerando o(s) novo(s) componente(s) redundante(s). Após a construção do modelo, segue-se o fluxo dos passos já descritos (veja sequência na Figura 5.1) até que as métricas de dependabilidade e custos estejam dentro dos requisitos.

5.2 ARQUITETURA BASE DE POTÊNCIA DE DATA CENTER

Os estudos de caso realizados tiveram como base a infraestrutura de potência de *data center* apresentada na Figura 5.2. A partir desta infraestrutura, outras foram geradas e avaliadas conforme o interesse em se obter cada métrica e a necessidade de cada estudo de caso.

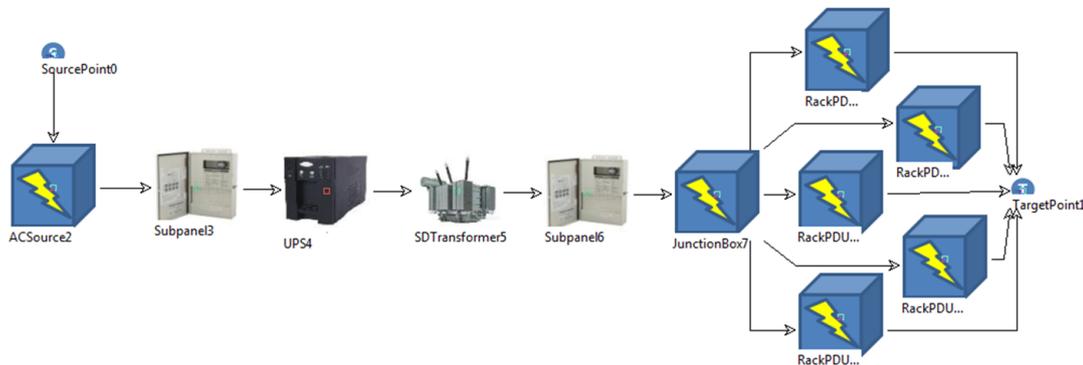


Figura 5.2 Arquitetura de potência base

Na Figura 5.2 a energia sai da *Fonte AC* (*ACSource2*), passando pelo painel de baixa voltagem (*Subpanel3*), pela *uninterruptible power supplies* (*UPS4*), pelas unidades de distribuição de energia (composto por um transformador (*SDTransformer5*) e um subpainel (*Subpanel6*)), por caixas de junção (*Junctionbox7*) e, finalmente, para as unidades de distribuição de energia para os dispositivos de TI (*Rack PDUs*).

O *AC source* é a fonte de energia, ou seja, contém os tempos para falha e reparo correspondentes ao fornecimento de energia pela concessionária para o *data center*. O *sub panel* é um painel por onde passa a energia. UPS é um conjunto de baterias que fornecem energia para o *data center* em caso de falta de energia pela fonte AC. O transformador é responsável por estabilizar a energia que chega para deixá-la na potência e corrente corretas para os equipamentos. *Junction Box* são caixas onde os cabos elétricos são

distribuídos para os *racks*.

O sistema apresentado na Figura 5.2 é considerado operacional se pelo menos um (1) bloco *RackPDU* estiver funcionando e for suprido de energia. Nesta figura, cada bloco *RackPDU* alimenta 10 racks.

5.3 ESTUDO DE CASO I: *RI* APLICADA A INFRAESTRUTURAS DE POTÊNCIA

Neste estudo de caso, é considerado o índice tradicional de índice de importância para confiabilidade e é utilizada a metodologia apresentada na Seção 5.1. O índice é calculado para infraestruturas de potência de *data centers* e o tempo considerado para avaliação é de 8760 horas.

O objetivo deste estudo de caso é demonstrar a aplicação do índice de importância para a confiabilidade utilizando a ferramenta ASTRO. Com isso obter infraestruturas melhores (em termos de dependabilidade), aplicando redundância, baseado neste índice.

A partir da arquitetura base, que representa uma estrutura simples de potência (energia) para *data center*, são realizadas sucessivas análises de importância (*RI*) e dependabilidade para se propor uma infraestrutura melhor. A partir de cada arquitetura, uma nova arquitetura é proposta (baseada no índice de importância).

A Figura 5.3 apresenta todas as arquiteturas analisadas, onde a Figura 5.3(a) é a Arquitetura Base (nomeada de Arquitetura A1) e as outras figuras correspondem às arquiteturas geradas. A Tabela 5.1 apresenta os valores para o índice de importância para confiabilidade de cada arquitetura. Adicionalmente, as métricas resultantes da análise de dependabilidade são mostradas na Tabela 5.2.

Após analisar a dependabilidade e os custos da arquitetura base (Arquitetura A1), o índice de importância para confiabilidade dos componentes foi calculado. Na Tabela 5.1, na linha correspondente da Arquitetura A1, é possível observar que o componente *Fonte AC*, seguido pela UPS, correspondem aos componentes com maior impacto na confiabilidade do sistema. Isto reflete o fato de que, dentre todos os componentes que estão em série, estes possuem o menor valor de MTTF, possuindo, conseqüentemente, menor confiabilidade.

Assim, a partir da análise dos índices de importância calculados, a Arquitetura A2 (veja Figura 5.3(b)) é proposta. Esta arquitetura tem como base a Arquitetura A1, adicionado um componente *Fonte AC* redundante. Para considerar este componente (*Fonte AC*) é necessário adicionar um componente STS, que realiza o chaveamento entre as duas Fontes AC.

Baseado nos resultados da análise de importância da confiabilidade da Arquitetura A2 é possível notar que, mesmo com com uma *Fonte AC* redundante, este componentes aparece como mais importantes, pois possuem os maiores valores para o índice de importância da confiabilidade. Considerando que a instalação e manutenção de mais de duas Fontes AC ser muito elevada, e nem sempre haver outras redes disponíveis, a partir

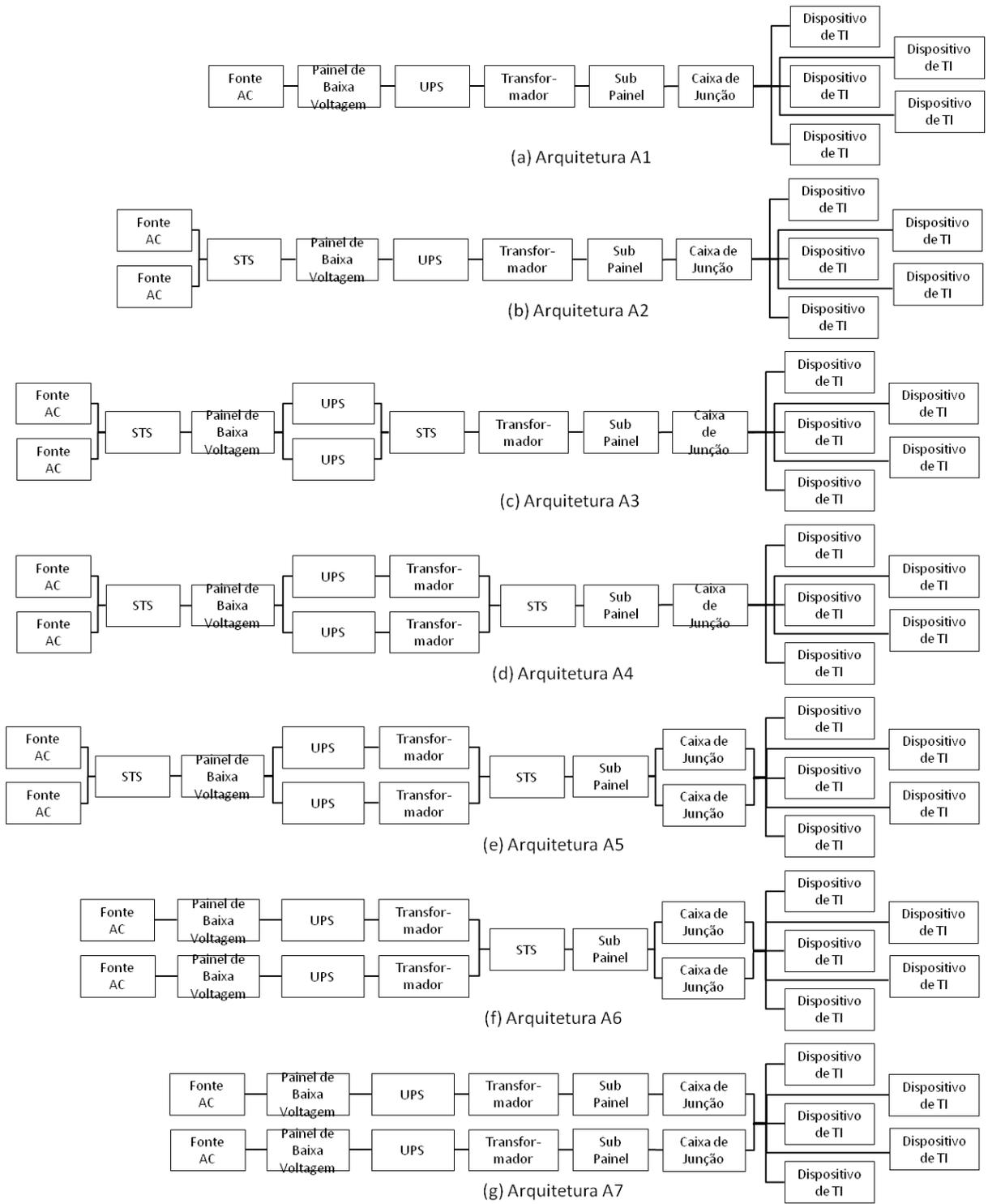


Figura 5.3 Arquiteturas de Potência de Data Center Consideradas

daqui será considerado próximo componente mais importante (que tenha o segundo maior valor na taxa).

Tabela 5.1 Índice de importância para confiabilidade para as arquiteturas A1 a A7

Arquitetura	Fonte AC	PBV	UPS	Transf.	Sub Painel	CJ	DTI	STS
A1	1,000	0,136	0,140	0,137	0,136	0,136	3,3E-08	
A2	1,000	0,293	0,302	0,295	0,293	0,294	7,2E-08	0,303
A3	1,000	0,293	0,010	0,295	0,293	0,294	7,2E-08	0,303
A4	1,000	0,293	0,013	0,013	0,293	0,294	7,2E-08	0,303
A5	1,000	0,293	0,013	0,013	0,293	0,002	7,2E-08	0,303
A6	1,000	0,136	0,140	0,137	0,292	0,002	7,2E-08	0,301
A7	1,000	0,136	0,140	0,137	0,136	0,136	7,2E-08	

Abreviaturas: PVB - Painel de Baixa Voltagem; UPS - *Uninterruptible Power Supply*; Transf. - Transformador; CJ - Caixa de Junção; DTI - Dispositivo de TI; STS - *Switch* de Transferência Estática;

Tabela 5.2 Valores para disponibilidade, confiabilidade (8760 horas) e custo das Arquiteturas de A1 a A7

Arquitetura	Disponibilidade (9s)	Confiabilidade (9s)	Custo (US\$)
A1	2,72528	0,05889	106.100,00
A2	4,01779	0,11239	121.900,00
A3	4,01204	0,11205	192.700,00
A4	4,06155	0,11342	193.250,00
A5	4,09470	0,11432	193.400,00
A6	4,37632	0,11376	192.800,00
A7	5,45056	0,11779	192.200,00

Desse modo, a Arquitetura A3 (conforme Figura 5.3(c)) é criada considerando a lista de importância de componentes da Arquitetura A2. Para isso, foi adicionado uma UPS redundante. O componente STS (segundo mais importante) não apresenta melhoria quando redundante, pois quando há 2 STS redundantes, em casos de oscilação forte de energia, ambos desligariam, e quando há chaveamento, ambos chaveiam para a mesma fonte. Na Arquitetura A3, é adicionado um novo STS, responsável por fazer o chaveamento entre as duas UPS.

A Arquitetura A4 foi proposta após avaliação de dependabilidade, custos e importância para confiabilidade da Arquitetura A3. Nesta nova arquitetura há um Transformador redundante (de acordo com os valores presentes na Tabela 5.1).

A Arquitetura A5 considera uma Caixa de Junção redundante, pois este é componente mais importante, a ser considerado, da Arquitetura A4 (veja Tabela 5.1).

Para gerar a Arquitetura A6, um componente Painel de Baixa Voltagem redundante é adicionado. Considerando que na Arquitetura A5 há dois componentes STS, a Arquitetura A6 pode ser criada de duas formas, mantendo os dois STS ou removendo um deles. A alternativa mais interessante é eliminar um STS (veja Figura 5.3(f)), uma vez que estamos interessados em dependabilidade e, quanto mais componentes conectados

em série, menor serão os índices (de dependabilidade).

Considerando a Arquitetura A6, o componente mais importante é o Sub Painel (conforme Tabela 5.1). Assim, a Arquitetura A7 considera a Arquitetura A6 com um novo SubPainel redundante. Adicionando este componente é possível eliminar o STS na Arquitetura A7, que não possui nenhum STS pois os equipamentos modernos de TI geralmente são *dual-corded* suportando duas fontes de alimentação paralelas.

A Figura 5.4 apresenta relação entre as sete arquiteturas considerando o aumento percentual para disponibilidade, confiabilidade e custo. Como esperado, a disponibilidade e a confiabilidade aumentam a medida que mais redundância é adicionada, exceto para a Arquitetura A3, onde foi adicionado um componente não-redundante. Além disso, a Arquitetura A7 é a opção mais interessante, pois a disponibilidade e a confiabilidade são as mais altas e tem um custo similar às outras arquiteturas (a partir da A3).

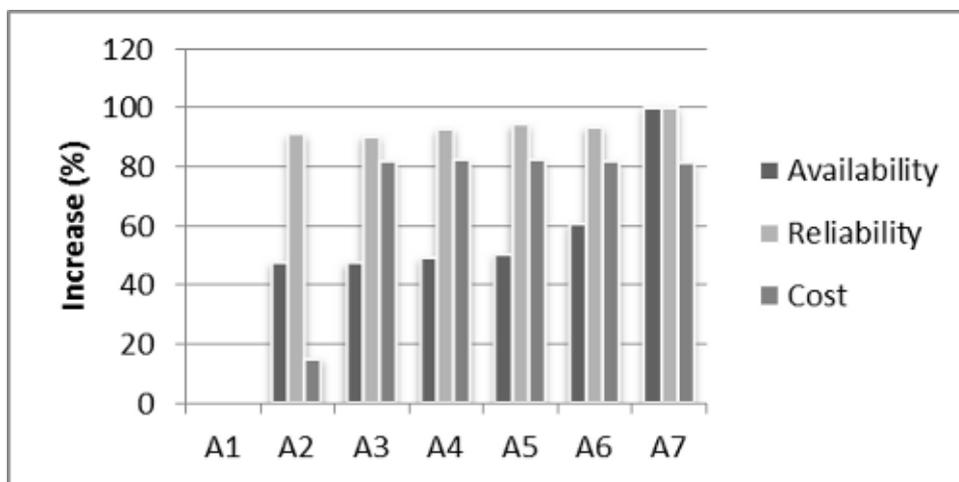


Figura 5.4 Increasing with redundancy considering RI

Outro caso interessante é da Arquitetura A2, pois comparando-a com a arquitetura base, tem um aumento na disponibilidade de aproximadamente 48% e na confiabilidade de aproximadamente 90%, com aumento nos custos próximo a 3%.

5.4 ESTUDO DE CASO II: RI_C APLICADA EM INFRAESTRUTURAS DE POTÊNCIA

Este estudo de caso apresenta a aplicação do índice de *importância da confiabilidade e custo* proposto na Seção 4.6.3, utilizando a metodologia apresentada na Seção 5.1. O índice é calculado para infraestruturas de potência de *data centers* e o tempo considerado para avaliação é de 8760 horas.

A Figura 5.5(a) apresenta a Arquitetura Base (infraestrutura de potência), a partir da qual foram propostas outras arquiteturas. Resumidamente, nesta arquitetura a energia sai da *Fonte AC*, passando pelo Painel de Baixa Voltagem, pela *uninterruptible power supplies* (UPS), pelas unidades de distribuição de energia (composto por um transformador e um

subpainel), por caixa de junção e, finalmente, pelas unidades de distribuição de energia para os Dispositivos de TI.

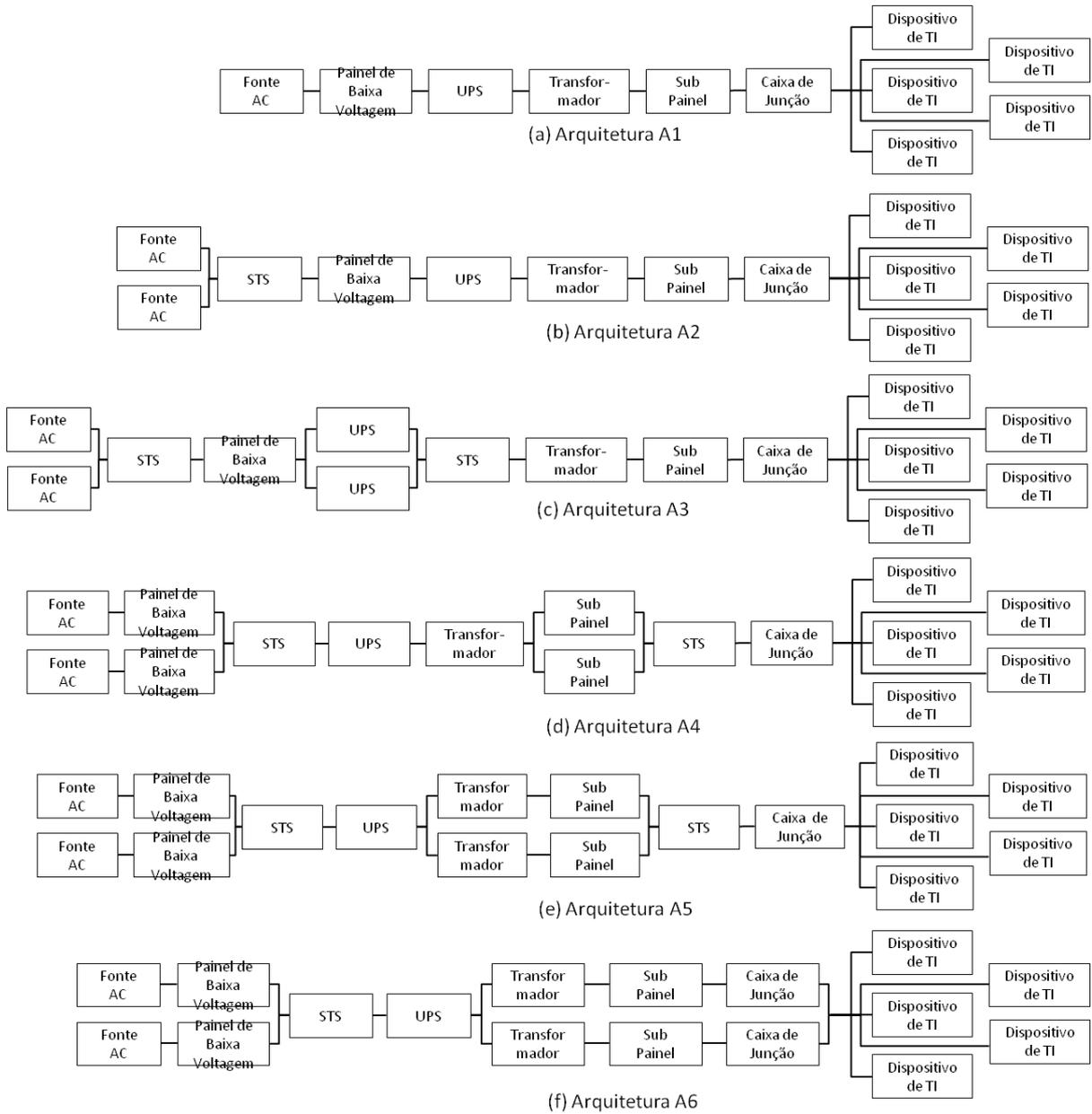


Figura 5.5 Arquiteturas de Potência de Data Center Consideradas

As métricas avaliadas são disponibilidade, confiabilidade (tempo de 8760 horas) e custo de aquisição. Cada nova arquitetura avaliada serve de base para propor uma nova arquitetura, assim 5 novas arquiteturas foram geradas (veja Figura 5.5) considerando a avaliação do índice de importância e custo. A Tabela 5.3 apresenta os valores para o índice de importância dos componentes das 6 arquiteturas.

Com o modelo construído, as métricas de dependabilidade e os custos da Arquitetura

Tabela 5.3 Índice de Importância de Componentes para Arquiteturas da Figura 5.5

	ACS1	ACS2	STS1	STS2	UPS1	SDT1	SDT2	JB1	JB2	SP1	SP2	LVP1	LVP2
A1	1				0,056	0,158		0,157		0,158		0,158	
A2	1	1	0,343		0,147	0,333		0,333		0,334		0,334	
A3	1	1	0,343		0,147	0,333		0,332		0,334		0,155	0,155
A4	1	1	0,342	0,342	0,148	0,333		0,332		0,002	0,002	0,155	0,155
A5	1	1	0,342	0,342	0,149	0,004	0,004	0,332		0,004	0,004	0,155	0,155
A6	1	1	0,342		0,148	0,004	0,004	0,004	0,004	0,004	0,004	0,155	0,155

Abreviaturas: ACS - Fonte AC; LVP - Painel de Baixa Voltagem; STS - *Switch* de Transferência Estática; UPS - *Uninterruptible Power Supply*; SDT - Transformador; SP - Sub Painel; JB - Caixa de Junção; ITD - Dispositivo de TI;

Base (A1) foram avaliadas. Estes dados são a base para a construção da Arquitetura A2. Em seguida a avaliação de importância de componentes foi realizada. Nesta avaliação, o componente *Fonte AC* apresenta o maior valor para o RIC .

Para criar a Arquitetura A2, um componente *Fonte AC* redundante foi adicionado. Para ser possível considerar esta redundância na arquitetura, é necessária a adição de outro componente, chamado *Switch* de Transferência Estática (*Static Transfer Switch* - STS), que é responsável realizar o chaveamento entre uma Fonte AC e outra.

A métricas de dependabilidade (confiabilidade e disponibilidade) foram avaliadas para a Arquitetura A2, além do custo de aquisição. Em seguida, o índice de importância (RIC) foi obtido.

Observando o resultado da análise de importância da Arquitetura A2 na Tabela 5.3, é possível notar que, mesmo possuindo uma *Fonte AC* redundante, este componente continua sendo o mais importante (no contexto de confiabilidade e custo). No entanto, assim como no Estudo de Caso I, será considerado o próximo componente com maior índice de importância na Tabela 5.3.

A Arquitetura A3 foi criada a partir da Arquitetura A2, considerando a importância dos componentes (RIC). Nesta arquitetura, um componente *Painel de Baixa Voltagem* foi adicionado de forma redundante. A inclusão do componente pode ser feita de duas formas: a primeira é incluindo um STS para fazer o chaveamento entre os dois e a outra forma, neste caso, é necessário conectar cada Painel em uma Fonte AC, usando o STS já existente para fazer o chaveamento entre os Painéis. Assim, nesta nova arquitetura, o componente STS faz o chaveamento entre dois Painéis de Baixa Voltagem (LVP) e não entre as Fontes AC.

Após analisar dependabilidade, custo e importância de componente da Arquitetura A3, a Arquitetura A4 foi gerada. Esta arquitetura considera um Sub Painel redundante (de acordo com os valores do índice de importância apresentado na Tabela 5.3). Esta Arquitetura possui dois componentes STS, um que faz o chaveamento entre dois Painéis de Baixa Voltagem e outro que realiza o chaveamento entre dois Sub Painéis.

Para criar a Arquitetura A5, foi considerado a Arquitetura A4 com redundância aplicada ao Transformador (SDT). A análise do índice de importância da Arquitetura A4

mostra que os componentes STS têm o maior valor de importância, mas como dito anteriormente (Estudo de Caso I), aplicar redundância no STS não é apropriado, assim, a redundância é aplicada ao próximo componente com o maior valor de importância (componente Transformador).

Finalmente, a Arquitetura A6 é concebida com redundância aplicada a Caixa de Junção. Nesta arquitetura é possível eliminar o STS que realiza o chaveamento entre os Sub Painéis e conectá-los diretamente às caixas de junção.

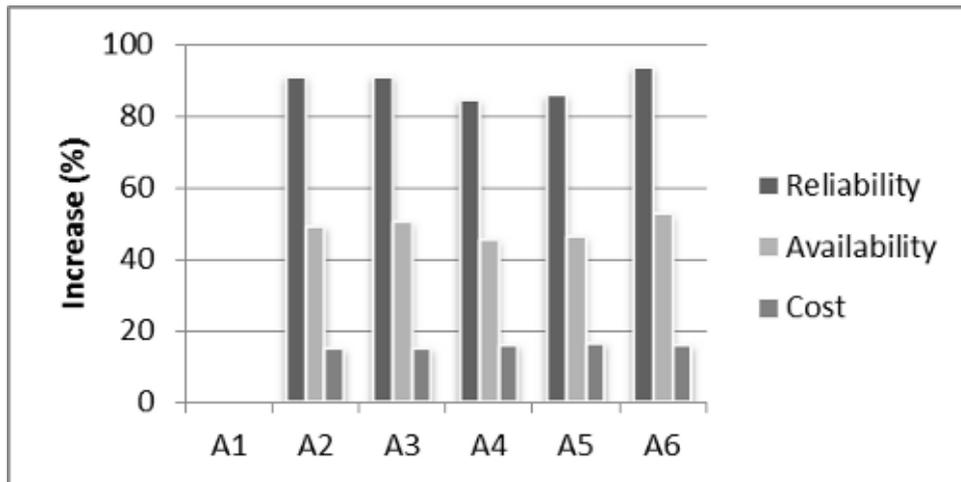


Figura 5.6 Comparação das Arquiteturas Geradas com a Base

A Tabela 5.4 apresenta os resultados para disponibilidade e confiabilidade (para 8760 horas) das arquiteturas, além do Custo Total de Aquisição. Adicionalmente, a Figura 5.6 apresenta o gráfico que relaciona os resultados. É possível observar que a Arquitetura A6 é uma opção interessante, uma vez que tem os maiores valores para confiabilidade e disponibilidade e estes resultados foram alcançados considerando um investimento de 15%, comparando-se com a arquitetura base.

Tabela 5.4 Valores de Disponibilidade, Confiabilidade e Custo para as Arquiteturas A1 a A6

Architecture	Availability (9s)	Reliability (9s)	Cost (US\$)
A1	2,72779	0,059651	106.100,00
A2	4,07010	0,113927	121.900,00
A3	4,09772	0,113981	122.100,00
A4	3,96712	0,110057	123.100,00
A5	3,99053	0,110823	123.650,00
A6	4,16355	0,115577	123.000,00

5.5 ESTUDO DE CASO III: AVALIAÇÃO DE DEPENDABILIDADE POR LIMITES

Nesse estudo de caso, três arquiteturas de potência para ambientes *data center* são analisadas. Estimativas de disponibilidade e confiabilidade, através da avaliação por limites, são obtidas com valores aproximados. As arquiteturas para este estudo de caso foram obtidas de [Callou et al.,].

As infraestruturas de potência são responsáveis pelo fornecimento ininterrupto de energia, estabilizada, aos dispositivos que fazem parte da infraestrutura de refrigeração, bem como para os dispositivos de TI. Alguns dispositivos podem ser replicados, a fim de aumentar a disponibilidade e confiabilidade da arquitetura de potência.

A Figura 5.7 apresenta a arquitetura base (Arquitetura A1) que corresponde a uma infraestrutura de potência com equipamentos redundantes, onde dois caminhos (*path1* e *path2*) estão disponíveis neste sistema. Neste sistema, a energia é fornecida por uma fonte de alimentação (*AC Source*), passando em seguida, por painéis de baixa voltagem, unidades de suprimento ininterrupto de energia (UPS), unidades de distribuição de energia (PDUs) (composto de transformador e painéis elétricos), caixas de junção e, finalmente, pelos os *racks* PDU (unidades de distribuição de energia para os *racks*).

Em relação a arquitetura base, considera-se que uma falha ocorre na infraestrutura de potência (e, conseqüentemente, no sistema) quando a falha dos equipamentos da Figura 5.7 impedem o fornecimento de energia para os componentes de TI (*IT Device*).

Além da arquitetura base, duas outras arquiteturas são consideradas, ambas maior redundância de equipamentos em relação a Arquitetura A1. A Arquitetura A2 (veja Figura 5.8(a)) possui um transformador redundante para cada caminho (*path*) de A1. Na Arquitetura A3 (ver Figura 5.8(b)), além dos transformadores, são adicionados dois componentes STS para possibilitar que ambas as fontes (*ACSource*) forneçam energia para qualquer um dos caminhos (*path1* e *path2*). Desse modo quando ocorre uma falha em uma fonte de alimentação, a outra é capaz de fornecer energia.

A Tabela 5.5 apresenta os valores adotados de MTTFs e MTTRs nos dispositivos de potência (*fonte AC*, gerador, TI *Rack*, caixa de junção, STS, subpainel, transformador, UPS e painel de alimentação). O tempo utilizado para a realização da análise a confiabilidade foi de 8760 horas (1 ano) e a disponibilidade foi calculada para o estado estacionário.

Os resultados foram calculados para as iterações em todas as arquiteturas. Na Tabela 5.6, para efeitos de demonstração, é apresentado o conjunto parcial de valores da Arquitetura A1. Por exemplo, são apresentados os resultados para cada 3 iterações, totalizando 13 valores para o limite superior. Nessa abordagem, para cada novo valor apresentado, mais três passos (ou iterações) são apresentados. Todos os passos foram incluídos, desse modo, o ultimo valor obtido corresponde ao valor exato da métrica.

A Figura 5.9 apresenta os gráficos gerados pela ferramenta que destacam as aproximações ao valor exato da métrica tanto para os limites superiores quanto inferiores da

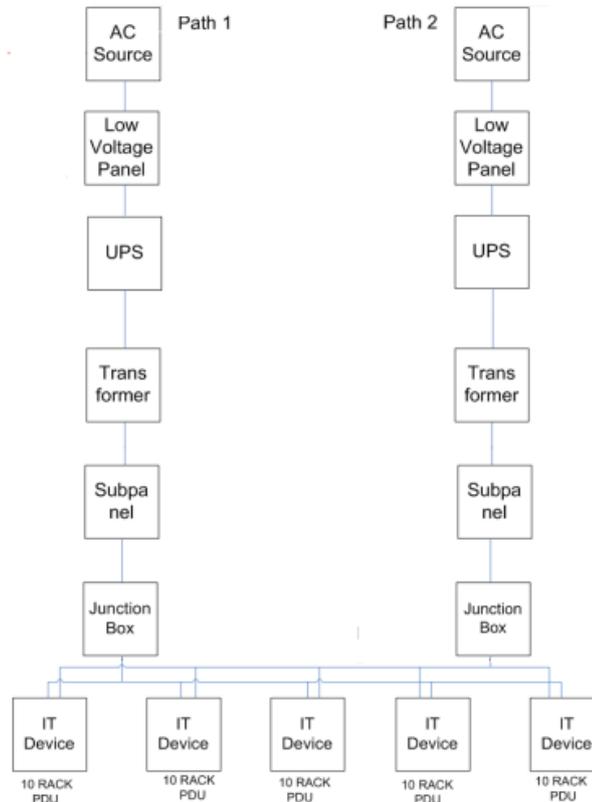


Figura 5.7 Infraestrutura com dois sistemas de potência redundantes

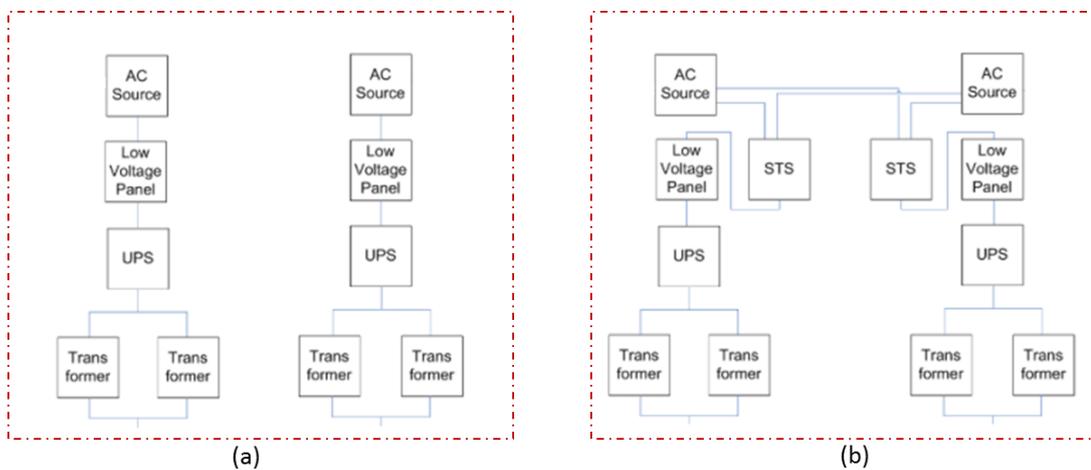


Figura 5.8 Infraestruturas de potência A2 (a) e A3 (b)

Arquitetura A1. Os gráficos foram gerados a partir dos valores de disponibilidade (*Steady State Availability*) e confiabilidade (*Reliability*). Na figura é possível observar a variação dos valores das métricas conforme mais passos são calculados, até a obtenção do valor exato. A diferença entre o limite superior e inferior vai sendo reduzida até chegar a um

Tabela 5.5 Parâmetros para as Arquiteturas A1, A2 e A3

Equipamento	MTTF (hs)	MTTR (hs)
Fonte AC	4,380	8
Gerador	2,190	8
TI Rack	378,148,937	4
Caixa de junção	26,120,000	8
STS	240,384	8
Subpainel	1,520,000	8
Transformador	763,201	8
UPS	250,000	8
Painel de alimentação	1,520,000	8

Tabela 5.6 Resultado da Avaliação dos Limites Superiores para A1 (a cada 3 passos)

Passo	Confiabilidade (9's)	Disponibilidade (9's)
1 ^o passo	6,9491	13,0278
4 ^o passo	4,8096	10,8781
7 ^o passo	3,5313	9,2405
10 ^o passo	3,2534	9,0965
13 ^o passo	2,2756	7,9819
16 ^o passo	2,2531	7,9719
19 ^o passo	1,9720	7,6772
22 ^o passo	1,9127	7,6497
25 ^o passo	1,3947	7,0920
28 ^o passo	1,3920	7,0908
31 ^o passo	1,3434	7,0395
34 ^o passo	1,0810	6,7707
37 ^o passo	0,1193	5,4556

ponto em comum no eixo das ordenadas (valor exato da métrica). Esta redução pode ser acentuada para determinados passos, dependendo da estrutura do modelo.

Para comparação das arquiteturas, foram considerados o primeiro passo, em seguida aquele que corresponde a 50% dos passos executados e, finalmente, o último passo, obtendo o valor exato para métrica. A Tabela 5.7 apresenta os três valores obtidos para as métricas nos passos informados (1^o, 50% e 100% do total de passos) para as Arquiteturas A1, A2 e A3. Os valores apresentados são para as métricas de disponibilidade e confiabilidade.

As Figuras 5.10 e 5.11, apresentam gráficos com a comparação das métricas obtidas através da análise por limites. Nestes gráficos é mostrada a relação entre as Arquiteturas A1, A2 e A3, para as métricas de Disponibilidade (Figura 5.10) e Confiabilidade (Figura 5.11).

Nas Figuras 5.10 e 5.11 são apresentados os limites (superiores e inferiores) obtidos

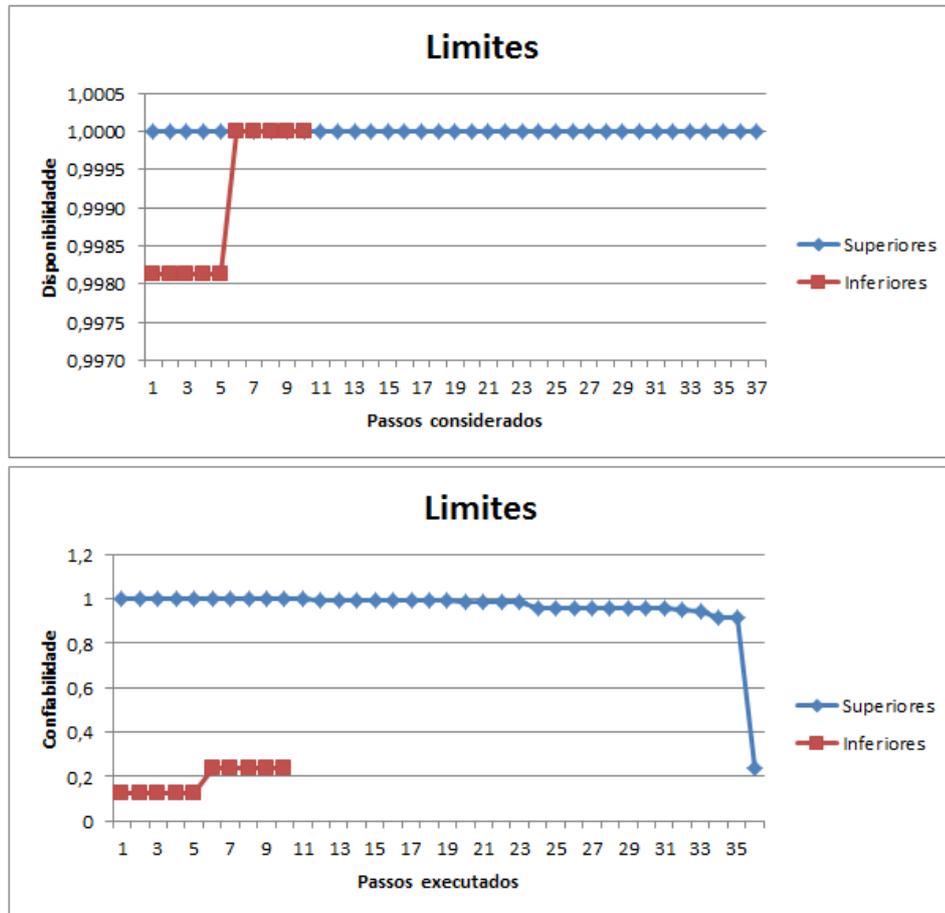


Figura 5.9 Limites para Disponibilidade e Confiabilidade da Arquitetura A1

Tabela 5.7 Limites para as arquiteturas A1, A2 e A3

	Passos	Confiabilidade (9's)	Disponibilidade (9's)
A1	1 ^o	[0,05964 ; 6,94908]	[2,72779 ; 13,02784]
	50%	[0,05965 ; 1,97200]	[2,72780 ; 7,67716]
	100%	[0,11930 ; 0,11930]	[5,45559 ; 5,45559]
A2	1 ^o	[0,05964 ; 6,94908]	[2,72779 ; 13,02783]
	50%	[0,06004 ; 1,45158]	[2,72910 ; 7,15289]
	100%	[0,12263 ; 0,12263]	[5,45688 ; 5,45688]
A3	1 ^o	[0,05736 ; 2,89258]	[2,72015 ; 8,95566]
	50%	[0,11473 ; 2,23704]	[4,10000 ; 8,28203]
	100%	[0,1245 ; 0,12456]	[5,47681 ; 5,47681]

com: (a) 1^o caminho e 1^o corte considerados, (b) 50% dos caminhos e cortes e, 100% significa que o valor foi obtido incluindo todos os caminhos e todos os cortes.

A Figura 5.10 e a Tabela 5.7 mostram que, com 50% das iterações calculadas para

a disponibilidade, a Arquitetura A3 apresenta um limite inferior (e um intervalo) acima daquele para as Arquiteturas A1 e A2. Isso indica que o valor exato da métrica pode ser superior ao das outras arquiteturas, de modo que a Arquitetura A3 pode ser a mais indicada em relação as métricas desejadas. Caso o objetivo do projetista seja, por exemplo, obter uma infraestrutura cuja disponibilidade não seja inferior e 4 noves (0,9999), com 50% dos passos para o limite inferior, já estaria garantido esta disponibilidade na Arquitetura A3, eliminando a necessidade de calcular o valor exato ou de calcular mais passos.

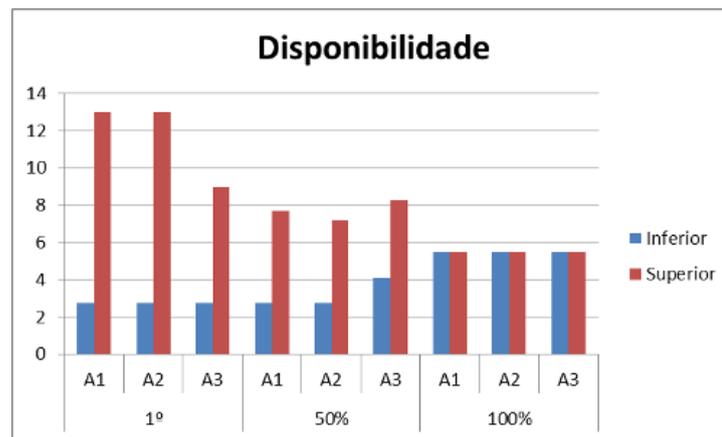


Figura 5.10 Comparação dos limites para Disponibilidade das Arquiteturas A1, A2 e A3

Analisando o gráfico da Figura 5.11 é possível notar que, com 50% das iterações (para confiabilidade) realizadas, na Arquitetura A3 já se obtém resultados que podem ser viáveis ao projetista, tanto para o limite inferior quanto para o limite superior, pois estão acima daqueles obtidos nas outras estruturas com a mesma proporção de iterações realizada.

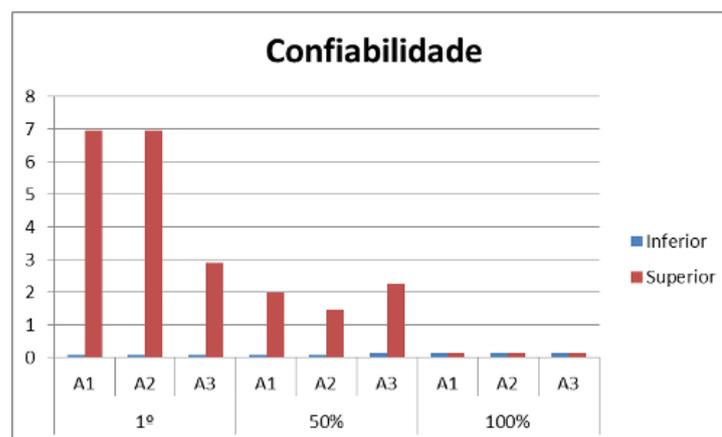


Figura 5.11 Comparação dos limites para Confiabilidade das Arquiteturas A1, A2 e A3

Baseado nas análises de 50% do total de iterações realizadas, a Arquitetura A3 apresenta os valores para a disponibilidade e para a confiabilidade superiores aos das outras

arquitecturas. Já com 100% das iterações, conforme Tabela 5.7, as métricas exatas da Arquitectura A3 são superiores às das outras arquitecturas, comprovando o comportamento que fora observado com 50% das iterações.

5.6 CONSIDERAÇÕES FINAIS

Neste Capítulo foram apresentados quatro estudos de caso para mostrar algumas das novas funcionalidades (extensões) da ferramenta Astro. No Estudo de Caso I, o tradicional índice de importância da confiabilidade foi aplicado em várias infraestruturas de potência de *data center*, identificando pontos fracos, para se obter infraestruturas mais confiáveis. Foram obtidos ainda, para as infraestruturas, os valores de confiabilidade, disponibilidade e TCO.

No segundo estudo de caso, seis arquitecturas para sistemas de potência, desta vez utilizando o índice de importância da confiabilidade e custo, foram analisadas, considerando recursos financeiros limitados. Os valores para disponibilidade, confiabilidade e custo de aquisição foram obtidos para cada uma das arquitecturas utilizando modelos RBD.

No terceiro estudo de caso foi aplicado o cálculo das métricas de dependabilidade por aproximações em três arquitecturas com diferentes níveis de redundância. Foi demonstrado, também, como se obter o valor exato das métricas utilizando a abordagem de limites (aproximação).

A partir destes estudos de caso é possível observar que a utilização das novas funcionalidades da ferramenta Astro (por exemplo análise de índices de importância) é possível avaliar sistemas *data center*. Além disso, a ferramenta é genérica o suficiente para avaliar sistemas em geral (por meio de RBD) considerando diferentes aspectos de dependabilidade como importância de componentes, custos e valores aproximados.

Aplicar redundância para melhorar a disponibilidade e a confiabilidade tem sido uma prática largamente adotada nos projetos de engenharia. Aplicar redundância visando melhorar (de algum modo) a estrutura, implica em utilização de componentes adicionais, exigindo mais investimento. Além disso, aplicar redundância pode, dada a especificidade dos componentes, causar sobrecarga ou subutilização de alguns equipamentos. Nos *data centers* não tem sido diferente, pois as infraestruturas precisam ser altamente confiáveis e estarem disponíveis 24 horas por dia, 7 dias por semana, necessitando de melhorias constantes.

Para ser possível alcançar estes requisitos, projetistas precisam de suporte adequado, que os permita selecionar, adotar ou construir soluções mais interessantes, considerando as métricas de dependabilidade e os custos envolvidos no projeto.

Assim, métricas específicas, que incluam aspectos nem sempre considerados, como custos e importância de componentes, são fundamentais e devem estar disponíveis para utilização. Além disso, é importante o apoio de ferramentas especializadas, que forneçam, além das métricas tradicionais de dependabilidade, novas métricas que possam ser utilizados paralelamente, guiando estes projetistas às melhores decisões.

Neste contexto, os índices de importância para a confiabilidade e custos, importância para a disponibilidade e custos foram criados para auxiliar projetistas e engenheiros nos projetos de infraestruturas de *data center*, mas podem ser adotados para infraestruturas em geral.

A ferramenta Astro, desenvolvida com foco em *data center* e apresentada por Silva [Silva et al., 2010], fornece um ambiente integrado para suporte para análise de dependabilidade e sustentabilidade. Esta ferramenta considera as vantagens de Redes de Petri Estocásticas (SPN) e Diagramas de Blocos para a Confiabilidade (RBD) que permitem a análise de modelos em geral.

A ferramenta possui quatro ambientes que são: avaliação RBD, avaliação SPN, avaliação de infra-estruturas de potência em sistemas *data center* e avaliação de infraestruturas de resfriamento em sistemas *data center*. Com o uso dos ambientes de *data center*, usuários não especializados em técnicas de modelagem SPN e RBD podem avaliar um determinado sistema *data center* conhecendo apenas sua estrutura. A ferramenta Astro, apesar de ter herdado várias funcionalidades da ferramenta Mercury e de implementar novas, não incluía funcionalidades que são importantes na análise de infraestruturas (de

data center), principalmente quando se trata de redundância, como importância de componentes.

Assim, este trabalho apresentou a criação de novos índices para auxiliar a análise de dependabilidade, incorporando estas métricas à ferramenta Astro, expandindo o ambiente RBD, visando auxiliar os projetistas em suas decisões. Estas incorporações vieram complementar o ambiente que faz uso de Diagramas de Bloco para a Confiabilidade (RBD), melhorando e ampliando a avaliação, incluindo aspectos antes não disponíveis nesta e em outras ferramentas.

6.1 CONTRIBUIÇÕES

As principais contribuições da pesquisa são, em linhas gerais, os novos índices propostos e as novas funcionalidades implementadas na ferramenta, que são descritos, de forma mais específica, a seguir:

Tratamento de Componentes Repetidos: inclusão do método para análise de modelos RBD que contenham componentes repetidos. Este método, além de aperfeiçoar as funções existentes na ferramenta, ampliou exponencialmente as possibilidades de modelagem para representação e análise de diversas estruturas, que anteriormente não eram passíveis de análise;

Análise de Dependabilidade por Aproximação: implementação do método para obtenção de valores aproximados para modelos RBD (Soma dos Produtos Disjuntos utilizando caminhos e cortes mínimos), representa uma alternativa ao método exato na análise de modelos grandes e complexos. Encontrar o valor exato pelo método padrão pode ser mais demorado que a obtenção de um valor aproximado. É possível, inclusive, calcular limites superiores e inferiores, obtendo-se um intervalo no qual está contido o valor exato da métrica de interesse;

Funções Estrutural e Lógica: os métodos para geração automática da função lógica e da função estrutural possibilitaram a visualização e análise da estrutura do modelo, por meio de expressões aritméticas e booleanas. Estas expressões podem ser analisadas e avaliadas a partir das informações dos estados dos componentes. Para estas funções é possível ainda realizar experimentos nos modelos considerando blocos falhos (para análise do estado do modelo), sem que seja necessário remover o componente para representar a falha;

Índices Tradicionais de Importância de Componentes: a implementação do índice de importância para a confiabilidade tornou possível a análise de modelos RBD para verificação de pontos fracos, auxiliando projetistas no processo de melhoria de infraestruturas, principalmente na aplicação de redundância, pois calcula os valores e os lista, indicando, em ordem decrescente, aqueles que representam gargalos no sistema.

Importância de Componentes x Custo: criação de índices que relacionam importância para a confiabilidade (ou disponibilidade) e custos de aquisição, permitindo quantificar e relacionar, impacto nas métricas de dependabilidade e nos custos. Estes índices indicam aos projetistas, os componentes que, quando se aplica redundância, representarão melhoria significativa na dependabilidade, com investimentos reduzidos, comparando-se a outros componentes. Estes índices estão disponíveis na ferramenta.

Simplificações do Modelo RBD: inclusão de simplificação de modelos RBD (na ferramenta) permitindo a obtenção de um modelo de alto nível, sem que seja necessário remodelar ou ter vários modelos representando cada uma das abstrações que fossem feitas. Ao invés disso, as estruturas mais internas são substituídas por um super componente que contém as informações referentes ao submodelo, garantindo a coerência e integridade das análises.

Por fim, quatro estudos de caso foram apresentados neste trabalho para demonstrar a viabilidade das novas funcionalidades implementadas na ferramenta e a aplicabilidade dos índices propostos. Nesses estudos de caso, mais de 10 arquiteturas foram avaliadas em termos de dependabilidade, custos e importância de componentes. Foi evidenciado que, com o uso da ferramenta, projetistas têm a possibilidade de escolher, de acordo com as suas restrições e especificações, qual arquitetura de *data center* melhor se adequa às necessidades, sejam elas custos, disponibilidade ou confiabilidade.

6.2 RESTRIÇÕES E TRABALHOS FUTUROS

Durante o desenvolvimento deste trabalho, algumas limitações da ferramenta e dos índices propostos foram verificadas. Estas limitações são descritas abaixo, além de representarem boas opções para trabalhos futuros:

Inclusão de outros custos relacionados: os índices de importância propostos e implementados não estimam custos de operação, violação de SLA ou de manutenção. Assim, como trabalhos futuros, podem ser criados índices que relacionem (todos) estes custos. Estes índices poderiam ser disponibilizados na ferramenta.

Importância de componentes para a sustentabilidade: são estimados os impactos na sustentabilidade para as arquiteturas (em termos de TCO), mas quando se trata de importância de componentes, não são calculados índices que relacionem a participação individual dos componentes no impacto global para a sustentabilidade.

Inclusão de outros modelos para dependabilidade: apesar da ferramenta dispor de dois dos principais modelos para análise de dependabilidade, novos modelos podem ser incluídos, dando mais liberdade ao projetista ou analista. Exemplo destes modelos são árvore de falta e cadeias de Markov.

Inclusão de outros índices clássicos de importância: o principal índice clássico de análise de importância de componentes está disponível na ferramenta, porém, índices como Importância Estrutural e Importância Crítica, podem ser incluídos, para auxiliarem no processo de análise.

Implementação de outros métodos de aproximação: outros métodos de aproximação, como o da Inclusão/Exclusão ou outros, poderiam ser implementados, fornecendo alternativas que podem ter desempenho superior.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Abd-Allah 1997] Abd-Allah, A. (1997). Extending reliability block diagrams to software architectures. *System*, 97:80–93.
- [Ajmone Marsan et al., 1984] Ajmone Marsan, M., Conte, G., and Balbo, G. (1984). A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comput. Syst.*, 2:93–122.
- [Alger 2005] Alger, D. (2005). *Build the best data center facility for your business*. Networking Technology Series. Cisco.
- [Armbrust et al., 2009] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, UC Berkeley Reliable Adaptive Distributed Systems Laboratory.
- [Armstrong 1997] Armstrong, M. (1997). Reliability-importance and dual failure-mode components. *IEEE Transactions on Reliability*, 46(2):212–221.
- [Arregoces e Portolani, 2003] Arregoces, M. and Portolani, M. (2003). *Data center fundamentals*. Cisco Press.
- [Augarten 1984] Augarten, S. (1984). *Bit by Bit: An Illustrated History of Computers*. Houghton Mifflin Co. Boston, MA, USA.
- [Avizienis et al., 2001] Avizienis, A., Laprie, J., Randell, B., and of Newcastle upon Tyne. Computing Science, U. (2001). Fundamental concepts of dependability. *TECHNICAL REPORT SERIES-UNIVERSITY OF NEWCASTLE UPON TYNE COMPUTING SCIENCE*.
- [Balbo e Chiola, 1989] Balbo, G. and Chiola, G. (1989). Stochastic Petri net simulation. In *Proceedings of the 21st conference on Winter simulation*, pages 266–276. ACM.
- [Bash et al., 2008] Bash, C., Patel, C., Shah, A., and Sharma, R. (2008). The sustainable information technology ecosystem. In *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. IThERM 2008. 11th Intersociety Conference on*, pages 1126–1131. IEEE.
- [Basili et al., 2004] Basili, V., Donzelli, P., and Asgari, S. (2004). A unified model of dependability: Capturing dependability in context. *Software, IEEE*, 21(6):19–25.

- [Baumann e Tillman, 2004] Baumann, H. and Tillman, A. (2004). *The Hitch Hiker 's Guide to LCA*. Studentlitteratur.
- [Bause e Kritzinger, 2002] Bause, F. and Kritzinger, P. (2002). *Stochastic Petri Nets*. Vieweg.
- [Bautista e Sharma, 2007] Bautista, L. and Sharma, R. (2007). Analysis of environmental data in datacenters. Technical report, Technical report, HP Labs.
- [Birnbaum 1969] Birnbaum, Z. (1969). On the importance of different components in a multicomponent system. Technical report, DTIC Document.
- [Callou et al.,] Callou, G., Maciel, P., Magnani, F., Tavares, E. Sousa, E., Silva, B., Figueirêdo, J. J. C., Araujo, C., and Neves, F. Sustainability and Dependability Evaluation on Data Center Architectures. In *Systems Man and Cybernetics (SMC), 2011 IEEE International Conference on*.
- [Callou et al., 2010] Callou, G., Sousa, E., Maciel, P., Tavares, E., Araujo, C., Silva, B., Rosa, N., Marwah, M., Sharma, R., Shah, A., et al. (2010). Impact analysis of maintenance policies on data center power infrastructure. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 526–533. IEEE.
- [Chang et al., 2004] Chang, Y., Amari, S., and Kuo, S. (2004). Computing system failure frequencies and reliability importance measures using OBDD. *IEEE Transactions on Computers*, pages 54–68.
- [Chiola et al., 1995] Chiola, G., Franceschinis, G., Gaeta, R., and Ribaudó, M. (1995). GreatSPN 1.7: graphical editor and analyzer for timed and stochastic Petri nets. *Performance Evaluation*, 24(1-2):47–68.
- [Ciardo et al., 1989] Ciardo, G., Muppala, J., and Trivedi, K. (1989). SPNP: stochastic Petri net package. In *Petri Nets and Performance Models, 1989. PNPM89., Proceedings of the Third International Workshop on*, pages 142–151. IEEE.
- [Corporation 2003] Corporation, R. (2003). *System Analysis Reference: Reliability Availability and Optimization*. ReliaSoft Publishing.
- [Dhillon 2002] Dhillon, B. (2002). *Engineering maintenance: a modern approach*. CRC.
- [Dincer e Rosen, 2007] Dincer, I. and Rosen, M. (2007). *Exergy: Energy, Environment and Sustainable Development*. Elsevier Science.
- [Distefano et al., 2006] Distefano, S., Scarpa, M., and Puliafito, A. (2006). Modeling distributed computing system reliability with DRBD. In *25th IEEE Symposium on Reliable Distributed Systems, 2006. SRDS'06*, pages 106–118.
- [EMC e Lastemc, 2010] EMC and Lastemc (2010). *Information Storage and Management: Storing, Managing, and Protecting Digital Information*. Wiley.

- [Feria e Nunn, 2006] Feria, J. and Nunn, S. (2006). Method for determining total cost of ownership. US Patent 7,020,621.
- [Florin e Natkin, 1979] Florin, G. and Natkin, S. (1979). Matrix product form solution for closed synchronized queuing networks. In *Petri Nets and Performance Models, 1979. PNPM79., Proceedings of the Third International Workshop on*, pages 29–37. IEEE.
- [German et al., 1995] German, R., Kelling, C., Zimmermann, A., and Hommel, G. (1995). TimeNET: a toolkit for evaluating non-Markovian stochastic Petri nets. *Performance Evaluation*, 24(1-2):69–87.
- [Hilber e Bertling, 2007] Hilber, P. and Bertling, L. (2007). Component reliability importance indices for electrical networks. In *The 8th International Power Engineering Conference (IPEC)*.
- [Hirel et al., 2000] Hirel, C., Tuffin, B., and Trivedi, K. (2000). SPNP: Stochastic Petri Nets. Version 6.0. *Computer Performance Evaluation. Modelling Techniques and Tools*, pages 354–357.
- [IEEEGoldBook 1997] IEEEGoldBook (1997). *IEEE Recommended Practice for the Design of Reliable Industrial and Commercial Power Systems (The Gold Book)*. IEEE.
- [Kuo e Zuo, 2003] Kuo, W. and Zuo, M. (2003). *Optimal reliability modeling: principles and applications*. Wiley.
- [Laprie et al., 1992] Laprie, J., Avizienis, A., and Kopetz, H. (1992). *Dependability: Basic concepts and terminology*. Springer-Verlag New York, Inc. Secaucus, NJ, USA.
- [Leemis 1995] Leemis, L. (1995). *Probabilistic Models and Statistical Methods*. Prentice Hall.
- [Leitner et al., 2010] Leitner, P., Michlmayr, A., Rosenberg, F., and Dustdar, S. (2010). Monitoring, prediction and prevention of sla violations in composite services. In *2010 IEEE International Conference on Web Services*, pages 369–376. IEEE.
- [Li et al., 2010] Li, H., Casale, G., and Ellahi, T. (2010). Sla-driven planning and optimization of enterprise applications. In *Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering*, pages 117–128. ACM.
- [Maciel et al., 2010] Maciel, P. R. M., Trivedi, K., Mathias JR, R., and Kim, D. (2010). *Dependability Modeling In: Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*. Ed. Hershey: IGI Global, Pennsylvania, USA.
- [Marwah et al., 2010a] Marwah, M., Maciel, P., Shah, A., Sharma, R., Christian, T., Almeida, V., Araújo, C., Souza, E., Callou, G., Silva, B., et al. (2010a). Quantifying the sustainability impact of data center availability. *ACM SIGMETRICS Performance Evaluation Review*, 37(4):64–68.

- [Marwah et al., 2010b] Marwah, M., Maciel, P., Shah, A., Sharma, R., Christian, T., Almeida, V., Araújo, C., Souza, E., Callou, G., Silva, B., Galdino, S., and Pires, J. (2010b). Quantifying the sustainability impact of data center availability. *SIGMETRICS Perform. Eval. Rev.*, 37:64–68.
- [Marwah et al., 2009] Marwah, M., Sharma, R., Shih, R., Patel, C., Bhatia, V., Mekana-purath, M., Velumani, R., and Velayudhan, S. (2009). Data analysis, visualization and knowledge discovery in sustainable data centers. In *Proceedings of the 2nd Bangalore Annual Compute Conference on 2nd Bangalore Annual Compute Conference*, pages 1–8. ACM.
- [Mckinley et al., 2006] Mckinley, P. K., Samimi, F. A., Shapiro, J. K., and Tang, C. (2006). Service clouds: A distributed infrastructure for composing autonomic communication services. In *Proc. of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06)*, pages 341–348, Indianapolis, IN, USA.
- [Mihailescu et al., 2011] Mihailescu, M., Rodriguez, A., and Amza, C. (2011). Enhancing application robustness in infrastructure-as-a-service clouds. In *Proc. First International Workshop on Dependability of Clouds, Data Centers and Virtual Computing Environments (DCDV 2011) in conjunction with The 41st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2011)*, Hong Kong, China.
- [Molloy 1982] Molloy, M. K. (1982). Performance analysis using stochastic petri nets. *IEEE Trans. Comput.*, 31:913–917.
- [Paquette 2004] Paquette, L. (2004). The Sourcing Solution. *American Management Association, New York, NY*.
- [Patel et al., 2001] Patel, C., Bash, C., Belady, C., Stahl, L., and Sullivan, D. (2001). Computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications. In *ASME International Electronic Packaging Technical Conference and Exhibition (IPACK '01)*. Citeseer.
- [Patterson 2002] Patterson, D. (2002). A simple way to estimate the cost of downtime. In *Proc. 16th Systems Administration Conf. — LISA*, pages 185–8.
- [Rausand e Høyland, 2004] Rausand, M. and Høyland, A. (2004). *System reliability theory: models, statistical methods, and applications*. Wiley-IEEE.
- [ReliaSoft 2010] ReliaSoft (2010). BlockSim: System Reliability and Maintainability Analysis Software Tool. Disponível em <http://www.reliasoft.com/BlockSim/>.
- [Sahner e Trivedi, 1987] Sahner, R. and Trivedi, K. (1987). Reliability modeling using SHARPE. *Reliability, IEEE Transactions on*, 36(2):186–193.
- [Sahner et al., 1996a] Sahner, R., Trivedi, K., and Puliafito, A. (1996a). *Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package*. Kluwer Academic Pub.

- [Sahner et al., 1996b] Sahner, R. A., Trivedi, K. S., and Puliafito, A. (1996b). *Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package*. Kluwer Academic Publishers, Norwell, MA, USA.
- [Sathaye et al., 2000] Sathaye, A., Ramani, S., and Trivedi, K. (2000). Availability models in practice. In *FTCC-1. Proceedings of the Int. Workshop on Fault-Tolerant Control and Computing, Shoel, Korea*, pages 823–829.
- [Semaan 2008] Semaan, G. (2008). Designing networks with the optimal availability. In *Optical Fiber communication/National Fiber Optic Engineers Conference, 2008. OFC/NFOEC 2008. Conference on*, pages 1–6. IEEE.
- [Sharma et al., 2004] Sharma, R., Bash, C., Patel, C., and Beitelmal, M. (2004). Experimental investigation of design and performance of data centers. In *The Ninth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, 2004. IThERM'04*, pages 579–585.
- [Si et al., 2009] Si, S., Liu, F., and Cai, Z. (2009). Failure importance analysis models based on Bayesian network. In *Industrial Engineering and Engineering Management, 2009. IE&EM'09. 16th International Conference on*, pages 151–154. IEEE.
- [Si et al., 2010] Si, S., Zhang, L., Cai, Z., and Wang, N. (2010). Integrated importance measure of binary coherent systems. In *Industrial Engineering and Engineering Management (IE&EM), 2010 IEEE 17Th International Conference on*, pages 932–936. IEEE.
- [Silva et al., 2010] Silva, B., Maciel, P., Tavares, E., Araujo, C., Callou, G., Sousa, E., Rosa, N., Marwah, M., Sharma, R., Shah, A., et al. (2010). Astro: A tool for dependability evaluation of data center infrastructures. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 783–790. IEEE.
- [Stapelberg 2008] Stapelberg, R. (2008). *Handbook of reliability, availability, maintainability and safety in engineering design*. Springer Verlag.
- [Veras 2011] Veras, M. (2011). *Virtualização: Componente Central do Datacenter*. Brasport.
- [Vesely e Roberts, 1987] Vesely, W. and Roberts, N. (1987). *Fault tree handbook*. Nuclear Regulatory Commission.
- [Vismari] Vismari, L. Vigilância dependente automática no controle de tráfego aéreo: avaliação de risco baseada em modelagem em redes de Petri fluidas e estocásticas.(2007). Automatic dependent surveillance on air traffic control: risk assessment based on fluid stochastic Petri nets modeling.
- [Wang et al., 2004] Wang, W., Loman, J., and Vassiliou, P. (2004). Reliability importance of components in a complex system. In *Reliability and Maintainability, 2004 Annual Symposium-RAMS*, pages 6–11.

- [Wiboonrat 2008a] Wiboonrat, M. (2008a). An Optimal Data Center Availability and Investment Trade-Offs. In *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pages 712–719. IEEE.
- [Wiboonrat 2008b] Wiboonrat, M. (2008b). Risk anatomy of data center power distribution systems. In *Sustainable Energy Technologies, 2008. ICSET 2008. IEEE International Conference on*, pages 674–679. IEEE.
- [Xie et al., 2004] Xie, M., Dai, Y., and Poh, K. (2004). *Computing systems reliability: models and analysis*. Springer Us.
- [Zhang et al., 2006] Zhang, P., Portillo, L., and Kezunovic, M. (2006). Reliability and Component Importance Analysis of All-Digital Protection Systems. In *2006 IEEE PES Power Systems Conference and Exposition, 2006. PSCE'06*, pages 1380–1387.