



Pós-Graduação em Ciência da Computação

**“Um Algoritmo para Distribuição Balanceada
de Carga Elétrica e Redução de Consumo de
Energia em Centros de Dados e Nuvens”**

Por

JOÃO FERREIRA DA SILVA JÚNIOR

Dissertação de Mestrado



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, OUTUBRO/2013



Universidade Federal de Pernambuco

Centro de Informática

Pós-graduação em Ciência da Computação

João Ferreira da Silva Júnior

“Um Algoritmo para Distribuição Balanceada de Carga Elétrica e Redução de Consumo de Energia em Centros de Dados e Nuvens”

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

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

RECIFE, Agosto/2013

Catálogo na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Silva Júnior, João Ferreira da

Um algoritmo para distribuição balanceada de carga elétrica e redução de consumo de energia em centros de dados e nuvens / João Ferreira da Silva Júnior. - Recife: O Autor, 2013.

xiv, 78 f.: il., fig., tab.

Orientador: Paulo Romero Martins Maciel.

Dissertação (mestrado) - Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2013.

Inclui referências e apêndice.

1. Redes de computadores. 2. Sistemas distribuídos. 3. Dependabilidade. I. Maciel, Paulo Romero Martins (orientador). II. Título.

004.6

CDD (23. ed.)

MEI2013 – 130

Dissertação de Mestrado apresentada por **João Ferreira da Silva Júnior** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**Um Algoritmo para Distribuição Balanceada de Carga Elétrica e Redução de Consumo de Energia em Centros de Dados e Nuvens**” orientada pelo **Prof. Paulo Romero Martins Maciel** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Ricardo Massa Ferreira Lima
Centro de Informática / UFPE

Prof. Sérgio Mário Lins Galdino
Escola Politécnica de Pernambuco / UPE

Prof. Raimundo da Silva Barreto
Departamento de Ciência da Computação / UFAN

Prof. Paulo Romero Martins Maciel
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 28 de agosto de 2013

Profa. Edna Natividade da Silva Barros

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

Dedico essa dissertação a minha família, base de meu crescimento profissional, em especial: filho, esposa e mãe.

Agradecimentos

Agradeço a Deus e aos espíritos de luz por permitirem e favorecerem, de todas as formas (materiais e espirituais), a realização e conclusão deste trabalho. Por me iluminarem, guiarem e tornarem claras as decisões que tomei nesse período.

Ao meu orientador, Paulo Romero Martins Maciel, que confiou em minha capacidade, me incentivou e me acolheu, fazendo com que me tornasse um integrante do grupo de pesquisa MoDCS, agradeço-lhe pela oportunidade de crescimento profissional e intelectual. Tenho certeza que levarei um pouco de seu estilo de trabalho por toda a vida.

Ao amigo e parceiro desde o primeiro período da graduação, Gustavo Callou, por seu incentivo, compartilhamento de conhecimento e pelo trabalho desenvolvido em conjunto durante todo o período do mestrado, agradeço pela ajuda nas atividades acadêmicas e pela parceria nos poucos momentos de lazer que tivemos nesses últimos anos.

Ao colega Jorge Vieira, sou-lhe grato pelo suporte nas dúvidas de implementação dos algoritmos propostos. Aos colegas Geová Severo, Marinalva Praxedes e Romero Ferreira sou-lhes grato pelo apoio necessário ao início do curso.

Agradeço a Creusa Diniz, minha mãe, por seu apoio, orgulho e exemplos de moral e ética durante toda a vida. Agradeço a Simone Diniz, minha esposa e companheira de todas as horas, pelo incentivo, força, apoio, auxílio e por entender minhas infinitas horas ausentes de sua vida, ao longo dessa jornada. Agradeço também ao meu filho Kevin Diniz, por sua compreensão (mesmo que relutante por vezes) pelas várias horas dedicadas a este trabalho e ausência em seus alegres e contagiantes momentos de diversão.

Aos amigos que conquistei ao longo do curso: Jamilson, Rafael, Jean, Thiago, Rubens, Julian, Kadna, Érica, Renata, Danilo, Bruno Silva, Bruno Nogueira, Ermeson, Débora, Andresson, Hélio, Matheus, Lenin e tantos outros, agradeço pelas palavras de incentivo quando foi preciso, pelos momentos de dificuldades e ao mesmo tempo de descontração em meio a tanto trabalho e estudo, pelo auxílio quando precisei e, claro, pela amizade, que espero não acabar com o fim do curso de mestrado.

Agradeço aos professores Ricardo Massa, Sérgio Galdino e Raimundo Barreto, por aceitarem o convite para compor a banca de defesa do mestrado.

Finalmente, agradeço a todos aqueles, que de forma direta ou indireta contribuíram para a realização deste trabalho, que me incentivaram e apoiaram nesse período tão importante para a minha formação acadêmica.

*"É preferível saber poucas coisas muito bem a saber muitas coisas
muito mal."*

— CID CERCAL

Resumo

Os avanços tecnológicos acontecem por intermédio de pesquisas e estudos, inclusive com a criação de novos paradigmas, tais como a computação em nuvem, comércio eletrônico e redes sociais. Em alguns casos, grandes centros de dados são utilizados para suportar esses paradigmas. Com isso, estes grandes centros de dados tornaram-se elementos críticos no desenvolvimento de tarefas diárias, o que sugere que, a cada dia, utilizamos mais seus recursos, aumentando a demanda e, conseqüentemente, o consumo elétrico. Hodiernamente, o consumo de energia é uma questão de interesse comum. Pesquisas demonstram que, como consequência da constante evolução e expansão da tecnologia da informação, os centros de dados e as nuvens (*cloud computing*) são grandes consumidores de energia elétrica. Com esse alto consumo, destacam-se as questões de sustentabilidade e custo. Neste contexto, a presente dissertação propõe um algoritmo de distribuição de carga elétrica (ADCE) para otimizar a distribuição de energia em infraestruturas elétricas de centros de dados e nuvens privadas. O ADCE é suportado pelo ambiente denominado Mercury, que é capaz de realizar uma avaliação integrada das métricas de confiabilidade, custo e sustentabilidade. O ADCE otimiza, mais especificamente, a distribuição da corrente elétrica na modelagem de fluxo de energia (EFM - Energy Flow Model). A modelagem em EFM é responsável por estimar as questões de sustentabilidade e de custo das infraestruturas elétricas de centros de dados e nuvens, respeitando as restrições de capacidade de energia que cada dispositivo é capaz de fornecer (considerando o sistema elétrico) ou extrair (considerando o sistema de refrigeração). Ademais, dois estudos de caso são apresentados, onde são analisadas sete infraestruturas elétricas de um centro de dados e seis de uma nuvem privada. Para os centros de dados foi alcançada uma redução no consumo de energia de até 15,5% e a métrica utilizada para avaliar a redução do impacto ambiental (exergia) foi reduzida em mais da metade de seu valor. No que concerne ao estudo das nuvens privadas a redução do consumo de energia foi de até 9,7% e da exergia de para quase um terço de seu valor inicial. Os resultados obtidos foram significativos, uma vez que as preocupações ambientais ganham destaque com o passar dos anos e os recursos financeiros das empresas são finitos e muito valiosos.

Palavras-chave: ASTRO/Mercury; EFM - modelo de fluxo de energia; dependabilidade; sustentabilidade; arquiteturas elétricas de centro de dados ou nuvens privadas, ADCE - algoritmo para distribuição de carga elétrica; otimização.

Abstract

Technological advances happen through research and studies, including the creation of new paradigms such as cloud computing, e-commerce and social networking. In some cases, large data centers are used to support these paradigms. Thus, these data centers have become critical elements in the development of daily tasks, which suggests that, every day, we use more resources, increasing demand and hence the power consumption. Nowadays, energy consumption is an issue of common interest. Research demonstrates that as a consequence of the constantly evolving and expanding field of information technology, data centers and clouds are now major consumers of electrical energy. Such high electrical energy consumption emphasizes the issues of sustainability and cost. Against this background, the present paper proposes a power load distribution algorithm (PLDA) to optimize energy distribution of data center power infrastructures. The PLDA is supported by an environment called Mercury, capable of performing the integrated evaluation of dependability, cost and sustainability. More specifically, the PLDA optimizes the flow distribution of the energy flow model (EFM). EFMs are responsible for estimating sustainability and cost issues of data center and cloud infrastructures without crossing the restrictions of the power capacity that each device can provide (power system) or extract (cooling system). Additionally, two case studies are presented that analyzed seven data center and six private cloud power architectures. For data centers was achieved a reduction in energy consumption of up to 15.5 % and the metrics used to evaluate the reduction of environmental impact (exergy) was reduced by more than half of its value. On the study of private clouds the reduction of energy consumption was up 9.7 % and the exergy for nearly a third of its initial value. Significant results were observed, since environmental concerns are highlighted over the years and the financial resources of companies are finite and very precious.

Keywords: ASTRO/Mercury; EFM - energy flow model; dependability; sustainability; data center power architectures; private cloud power architecture; PLDA - power load distribution algorithm; optimization.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Abreviações	xiv
1 Introdução	1
1.1 Contexto	1
1.2 Motivação e Justificativa	4
1.3 Objetivos	5
1.4 Método Proposto	6
1.5 Trabalhos Relacionados	8
1.6 Organização da Dissertação	11
2 Fundamentação Teórica	12
2.1 Infraestrutura de Centro de Dados	12
2.2 Computação em Nuvem	15
2.3 Dependabilidade	16
2.3.1 Medida de importância para Componentes	20
2.4 Modelo de Dependabilidade	
Diagrama de Bloco de Confiabilidade	21
2.4.1 Função Estrutural	23
2.4.2 Função Lógica	24
2.5 Sustentabilidade	25
2.6 Modelagem de Fluxo Energético	27
2.6.1 Grafo Acíclico Dirigido	28
2.6.2 Custo	29

3	Algoritmo de Distribuição de Carga Elétrica	30
3.1	Contextualização	30
3.2	O Algoritmo	31
3.2.1	Caminho Válido	33
3.2.2	FASE I - Inicialização do ADCE	35
3.2.3	FASE II - Kernel do ADCE	36
3.2.4	FASE III - Distribuição de Pesos do ADCE	38
3.3	Exemplo 1	39
3.4	Exemplo 2	42
4	Estudos de Caso	45
4.1	Estudo de Caso 1 - Centro de Dados	45
4.1.1	Arquiteturas	48
4.1.2	Modelos	50
4.1.3	Resultados	52
4.2	Estudo de Caso 2 - Computação em Nuvem	55
4.2.1	Arquiteturas	57
4.2.2	Modelos	58
4.2.3	Resultados	61
4.3	Considerações Finais	63
5	Conclusão	64
5.1	Contribuições	66
5.2	Trabalhos Futuros	67
5.3	Considerações Finais	67
	Referências	69
	Appendices	75

Lista de Figuras

1.1	Visão geral do método aplicado	6
1.2	Exemplo de modelo SPN na ferramenta Mercury	7
2.1	Subsistemas de centro de dados	14
2.2	Exemplo de utilização da computação em nuvem	16
2.3	Arvore de Dependabilidade Adaptada de <i>Avizienis</i> [8]	18
2.4	(a) Blocos em Série; (b) Blocos em Paralelo	22
3.1	Parâmetros do modelo EFM	31
3.2	Parâmetros do modelo EFM	32
3.3	Exemplo de modelo EFM	33
3.4	Exemplo de modelo EFM - 1ª Iteração	34
3.5	Exemplo de modelo EFM - 2ª Iteração	34
3.6	Exemplo de modelo EFM - Final	35
3.7	Descrição da composição do Vetor A	38
3.8	Exemplo EFM	40
3.9	Exemplo EFM - Fase 2 concluída	41
3.10	Exemplo EFM - Concluído	42
3.11	Modelo EFM.	42
3.12	Modelo EFM com a execução do ADCE.	43
4.1	Arquiteturas elétricas de centros de dados A1 e A2	46
4.2	Arquiteturas elétricas de centros de dados A3 e A4	47
4.3	Arquiteturas elétricas de centros de dados A5, A6 e A7	48
4.4	Modelo RBD da Arquitetura A5.	50
4.5	Modelo EFM para Arquitetura A5	52

4.6	Melhora das arquiteturas com a aplicação do ADCE	54
4.7	Arquiteturas elétricas de nuvens privadas A1 e A2.	55
4.8	Arquiteturas elétricas de nuvens privadas A3 e A4.	56
4.9	Arquiteturas elétricas de nuvens privadas A5 e A6.	56
4.10	Arquitetura de nuvem privada	57
4.11	Arquitetura A4 na modelagem RBD.	59
4.12	Arquitetura A4 na modelagem EFM.	61
4.13	Melhora das métricas após aplicação do ADCE	62
A.1	Funcionalidades do Ambiente ASTRO/Mercury.	76
A.2	Exemplo do Ambiente Mercury.	77
A.3	Exemplo da modelagem EFM.	78

Lista de Tabelas

2.1	Valores de energia e exergia para diferentes modalidades energéticas. . .	26
3.1	Resumo dos resultados sem e com a execução ADCE.	43
4.1	Equipamento, Eficiência e Capacidade	49
4.2	Equipamentos e variáveis associadas - Estudo 1.	51
4.3	Relação entre equações, valores e variáveis.	52
4.4	Resultados da execução do ADCE em centro de dados com melhora em %	53
4.5	Descrição dos computadores utilizados na Nuvem	58
4.6	Capacidade e Eficiência	58
4.7	Valores de MTTF e MTTR	59
4.8	Relação entre equações, valores e variáveis - Estudo 2.	60
4.9	Resultados da execução do ADCE em nuvens com melhora em %. . . .	61

Lista de Abreviações

ADCE - Algoritmo de Distribuição de Carga Elétrica

EFM - Energy Flow Model ("Modelo de Fluxo Elétrico")

Eucalyptus - Elastic Utility Computing Architecture Linking Your Programs To Useful Systems

GAD - Grafo Acíclico Dirigido

HP - Hewlett-Packard

IaaS - Infrastructure as a Service ("Infraestrutura como um Serviço")

MC - Markov Chain ("Cadeia de Markov")

MoDCS - Modeling of Distributed and Concurrent Systems ("Modelagem de sistemas distribuídos e concorrentes")

MTTF - Mean Time to Fail ("Tempo médio de Falha")

MTTR - Mean Time to Repair ("Tempo médio de Reparo")

PaaS - Plataform as a Service ("Plataforma como um Serviço")

PFS - Priority First Search ("Busca de Primeira Prioridade")

RBD - Reliability Block Diagram ("Diagrama de Bloco de Confiabilidade")

RCI - Reliability and Cost Importance ("Importância da Confiabilidade e do Custo")

RI - Reliability Importance ("Importância da Confiabilidade")

SaaS - Software as a Service - ("Software como um Serviço")

SPN - Stochastic Petri Nets ("Redes de Petri Estocásticas")

TI - Tecnologia da Informação

TTF - Time to Fail ("Tempo de Falha")

TTR - Time to Repair ("Tempo de Reparo")

UPS - Uninterruptible Power Supply ("Fonte de Alimentação Ininterrupta")

1

Introdução

Este capítulo apresenta uma breve introdução, pertinente ao crescente consumo elétrico em proporções mundiais e à participação da tecnologia da informação nesse cenário. Além disso, propõe um algoritmo para redução do consumo elétrico em modelos de arquiteturas de centros de dados, destacando-se aspectos de dependabilidade, sustentabilidade e custos a serem avaliados de forma conjunta nesse contexto. Por fim, apresenta, ainda, as motivações, os objetivos, e a estrutura da presente dissertação.

1.1 Contexto

Com o desenvolvimento de novos paradigmas, tais como a computação em nuvem, o comércio eletrônico e as redes sociais, o consumo elétrico dos centros de dados, que fornecem as plataformas por eles utilizadas tem aumentado significativamente nos últimos anos em todo planeta [1] [3] [4]. Tal aumento de consumo elétrico se intensifica em decorrência do surgimento de grandes centros de dados, instalados para atender as necessidades imprescindíveis ao desenvolvimento de tarefas diárias, sugerindo, assim, que, a cada dia, sejam utilizados mais recursos computacionais.

Como exemplo, o *Google* e a *Microsoft* possuem uma imensa quantidade de recursos de TI, utilizados diariamente por todos. Estima-se que o primeiro possua mais de um milhão de servidores, enquanto que o segundo, apenas na cidade de Chicago, deve possuir mais de 300 mil [33]. Essas escalas tendem a aumentar significativamente com o passar dos anos, à medida que amadureçam as infraestruturas, plataformas e armazenamento [49] ou surjam novos paradigmas. Como as empresas competem entre si pela liderança deste segmento, a eficiência operacional de seus centros de dados assume uma importância

central para essas organizações [33].

Grandes centros de dados possuem uma enorme quantidade de servidores, o que implica em altos custos operacionais e de aquisição. Como exemplo, nos Estados Unidos os centros de dados consomem cerca de 1,5% de toda a energia produzida no país, o que representa um custo de \$4,5 bilhões de dólares ao ano, com expectativa de aumento com o passar dos anos [34]. Para a Europa Ocidental, há uma estimativa de aumento no consumo elétrico de 56 TWh, aferido em 2007, para 104 TWh, a ser alcançado em 2020 [59]. O consumo de energia dos centros de dados é um dos responsáveis por parte significativa desse aumento.

Paralelamente, a preocupação com o aumento do impacto ambiental tem assumido grande importância por parte da indústria, bem como da comunidade científica, motivadas pelo aumento do consumo elétrico e por diversas inquietações da sociedade, tais como mudanças climáticas, poluição e degradação ambiental. Por exemplo, as emissões de CO_2 podem aumentar e atingir valores entre 9% e 27% em 2030 [2]. Mas é importante lembrar que o dióxido de carbono produzido varia de acordo com a matéria prima utilizada na geração da energia: carvão, petróleo, gás natural e outros.

No Brasil, 70% a 93% da energia elétrica é proveniente de centrais hidrelétricas; uma forma bastante "limpa" (não agressiva ao meio ambiente) de geração de energia elétrica [47]. Nos Estados Unidos, a queima de carvão gera quase metade da eletricidade consumida. Entretanto emite 78% de todo o CO_2 produzido a partir de usinas de energia elétrica. Muitos países têm exigido a substituição de fontes de energia poluentes pelas não poluentes, tais como a energia solar, a eólica, a hidroelétrica, etc.

O crescimento da tecnologia da informação (TI), em conjunto com os novos paradigmas mencionados anteriormente, proporciona aumento da infraestrutura elétrica e de refrigeração dos centros de dados e, conseqüentemente, maior consumo de energia elétrica, face ao aumento da quantidade de componentes elétricos envolvidos diretamente com a disponibilidade do sistema. Assim, com uma preocupação financeira e sustentável por parte das empresas e visando auxiliar na tomada de decisão, é fundamental dispor de meios de avaliação dessas grandes estruturas em fase de projeto.

Algumas métricas importantes para avaliação de um centro de dados são utilizadas no presente trabalho, tais como confiabilidade, custo e sustentabilidade. Para computar essas métricas, foram utilizados os modelos: Reliability Block Diagram (RBD) [45], Stochastic Petri Nets (SPN) [57] e Energy Flow Model (EFM) [16] suportados pelo ambiente ASTRO/Mercury [56]. Os modelos RBD e SPN foram utilizados para se extrair

as métricas de dependabilidade dos sistemas avaliados, a fim de se obter subsídios para avaliação do modelo EFM, que propõe uma nova maneira para se calcular de forma integrada a sustentabilidade, a confiabilidade e os custos de um centro de dados [18] [15].

Este trabalho propõe um Algoritmo de Distribuição de Carga Elétrica (ADCE) para otimizar o fluxo de energia entre os componentes presentes no modelo EFM, considerando a eficiência elétrica dos dispositivos. O objetivo principal do algoritmo consiste na melhoria do fluxo de energia dentro do modelo quando existem caminhos redundantes, disponíveis para suportar a energia demandada. De fato, o algoritmo proposto visa aumentar a utilização desses caminhos que possuem níveis diferenciados de eficiência elétrica. Com a otimização do fluxo elétrico, várias métricas podem melhorar, como eficiência do sistema, sustentabilidade, custo operacional, entre outras.

É importante ressaltar que o EFM é suportado pelo ambiente ASTRO/Mercury. O ASTRO é uma ferramenta que analisa energia, refrigeração e infraestruturas de TI dos centros de dados. O kernel do ambiente, chamado Mercury, é responsável por realizar a avaliação dos modelos suportados (SPN, RBD, EFM e Cadeias de Markov - MC) [21]. O ASTRO possui visões para a modelagem de energia do centro de dados, refrigeração e infraestruturas de TI, a partir das quais os usuários não especializados podem avaliar a confiabilidade, a sustentabilidade e o custo, sem a necessidade de estarem familiarizados com os formalismos usados no Mercury. Modelos criados pelo ambiente ASTRO são automaticamente convertidos para modelos suportados pelo Mercury (para a avaliação de confiabilidade).

A avaliação do consumo elétrico dos centros de dados é efetuada por meio de métodos de aferição da sua eficiência energética. Considerando que a maioria das métricas encontradas foram desenvolvidas por organizações norte-americanas [47], é possível perceber uma relação de tais métricas com o conceito de centro de dados verde, uma vez que um menor consumo de energia elétrica significa menores níveis de emissão de gás carbônico na atmosfera, uma vez que, conforme acima mencionado, a matéria prima utilizada na geração de energia elétrica impacta na emissão de gás carbônico.

No Brasil, o conceito de centro de dados verde, de acordo com a análise de sua eficiência energética, não faz muito sentido, uma vez que a maior parte da energia elétrica gerada e consumida vem de fontes limpas. Por essa razão, a métrica utilizada para computar a eficiência energética neste trabalho foi a exergia, que é definida como “*a máxima fração de energia que pode teoricamente ser convertida em trabalho útil*” [54]. Em geral, no contexto das questões de sustentabilidade, a exergia é analisada em busca

de maximizar a eficiência do uso da energia ou avaliar a degradação de recursos naturais [60]. Nesta abordagem, busca-se por aferir a exergia consumida em um modelo EFM. Logo, quanto menor for o consumo de exergia no modelo, maior será o seu grau de sustentabilidade.

Entretanto, independentemente da métrica utilizada para fins de sustentabilidade, se houver a possibilidade de serem adotados procedimentos e técnicas para redução (ou otimização) do consumo de energia elétrica, que impliquem em uma economia de recursos financeiros (limitados e preciosos em qualquer organização), as iniciativas da computação verde também poderão ser aplicadas com sucesso no Brasil ou em qualquer outro país, independentemente da fonte geradora da energia.

Eis o objetivo deste trabalho. Para validar o algoritmo proposto, foram conduzidos dois estudos de caso, comparando a confiabilidade, sustentabilidade e os resultados dos custos, utilizando a configuração padrão do Mercury (onde há uma divisão do fluxo na rede, proporcional aos nós do modelo) e com a aplicação da técnica de otimização (ADCE). Esta medida foi utilizada para diferentes arquiteturas de energia de centros de dados e nuvens privadas.

1.2 Motivação e Justificativa

Notoriamente, observa-se ao longo dos últimos anos um crescimento considerável no consumo de energia em todo o planeta. Os centros de dados, compostos por equipamentos de TI, de infraestrutura elétrica e de refrigeração, são apontados como um dos responsáveis por grande parte desse consumo [47]. A gestão de energia e os impactos ambientais causados pelos centros de dados tornaram-se questões cada vez mais apreciadas pelas empresas, considerando o aumento dos custos de energia e a atenção global voltada para a sustentabilidade. Nos Estados Unidos, onde há muitos centros de dados, existe uma contribuição significativa para as emissões globais de carbono, uma vez que a matéria prima mais utilizada na geração da energia elétrica é o carvão mineral.

Ademais, as estimativas de consumo de energia dos centros de dados demonstram um aumento anual entre 4% e 8%. Como a tecnologia da informação, em todos os seus níveis, é responsável por uma fração considerável das emissões globais de carbono, uma pequena redução nos percentuais de consumo de energia dos centros de dados poderá acarretar em significativo impacto econômico e ambiental.

Destaca-se, em trabalhos anteriores, o desenvolvimento da ferramenta *ASTRO/Mercury* e de uma proposta de avaliação dos centros de dados, que tratam de forma integrada os conceitos de custo, disponibilidade e sustentabilidade. Tal ferramenta suporta modelos formais para avaliações de dependabilidade (RBD, SPN e MC), além de possuir uma visão capaz de permitir a modelagem da infraestrutura elétrica de um centro de dados (EFM). Entretanto, a elaboração do fluxo da rede elétrica permanece sob a responsabilidade do projetista e, conseqüentemente, sujeito a erros humanos ou sob a responsabilidade da distribuição padrão do Mercury. Dessa maneira, sem a precisão matemática necessária para assegurar uma boa resposta na modelagem elétrica dos centros de dados. Por essa razão, essa dissertação visa preencher as lacunas existentes nesse método de avaliação, fornecendo uma boa resposta (podendo ser ótima) para o consumo de energia de qualquer modelo de infraestrutura elétrica projetado na ferramenta *Mercury*, utilizando o modelo EFM.

1.3 Objetivos

Diante da sensibilidade da humanidade, quanto a agressões ao meio ambiente, manifestada em todas as partes do mundo, é crescente o sentimento de preocupação na redução do consumo de energia elétrica, bem como na adoção de estratégias mais sustentáveis para produção de eletricidade. Neste contexto, o presente trabalho propõe um algoritmo de distribuição do fluxo elétrico entre os componentes do modelo EFM de forma eficiente (otimizada), com relação ao consumo de energia. Em seguida, os objetivos serão relacionados, de forma específica:

- Propor uma heurística (baseada no consumo elétrico) para redução do consumo de energia elétrica em sistemas representados por modelos EFM.
- Desenvolver um algoritmo que produza um fluxo otimizado (baseado no consumo elétrico) entre os componentes do modelo EFM, de acordo com as características de cada equipamento.
- Integrar o algoritmo de otimização com os métodos de análise do EFM.
- Aplicar a estratégia desenvolvida a problemas de fluxo elétrico.

1.4 Método Proposto

Esta seção apresenta o método proposto para avaliar as infraestruturas de centros de dados e nuvens, considerando confiabilidade e questões de sustentabilidade. O método considera o ADCE como técnica de otimização para obtenção dos resultados a partir da avaliação dos modelos EFM.

A Figura 1.1 apresenta uma visão geral do método adotado. Sua primeira etapa trata sobre a análise do sistema, interfaces e interações. Nesta fase foi também conhecido o conjunto de métricas a serem avaliadas (disponibilidade, confiabilidade, custo, sustentabilidade e energia necessária), os modelos e suas propriedades (em particular RBD e EFM). Além da compreensão do tipo do problema a ser resolvido.

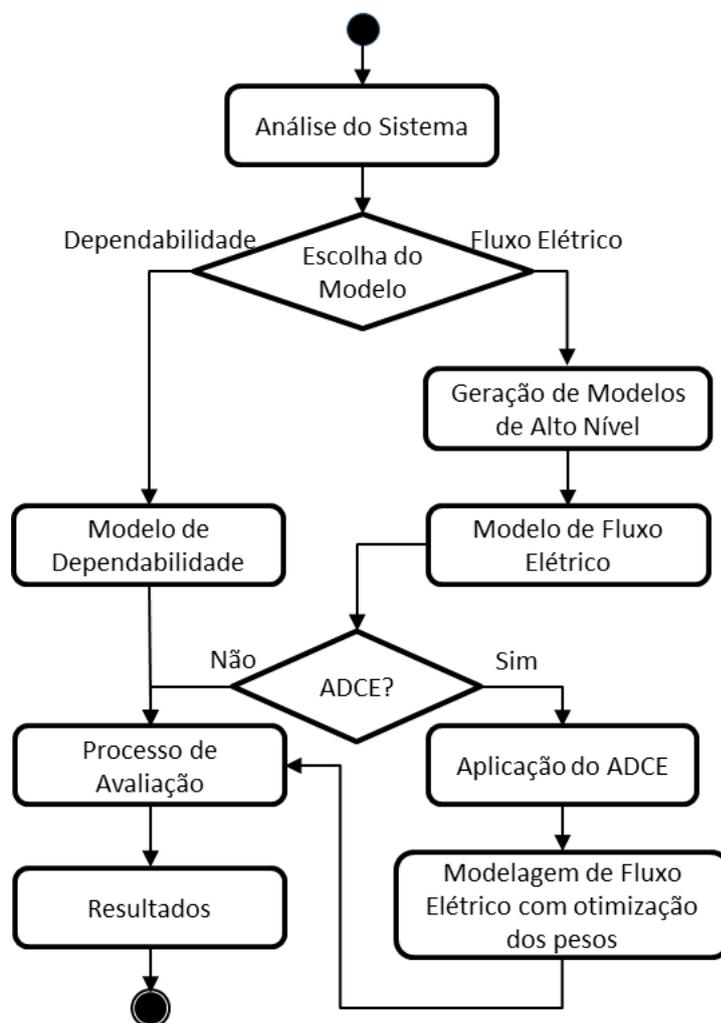


Figura 1.1: Visão geral do método aplicado

O próximo passo é a escolha do modelo a ser adotado para resolução de cada problema. A ferramenta utilizada foi o Mercury, pois nela há uma visão para vários tipos de modelos, onde se pode optar por MC, SPN, RBD ou EFM. Os problemas a serem resolvidos podem ser classificados como do tipo: avaliação da dependabilidade ou avaliação do fluxo de energia (considerando fluxo de resfriamento ou fluxo elétrico).

Caso avaliação de dependabilidade seja escolhida, é possível criar modelos SPN, RBD ou MC. A Figura 1.2 exibe um exemplo de um modelo SPN. Neste exemplo é possível perceber os lugares $P0$, $P1$, $P2$, $P3$, $P4$, $P5$, $P6$ e $P7$ com seus respectivos "tokens", as transições imediatas $T10$ e $T11$ e transições temporizadas $TE0$, $TE1$, $TE2$, $TE3$, $TE4$ e $TE5$. Além de um rótulo com o resultado do cálculo da confiabilidade desse sistema. Seguindo esse fluxo, o próximo passo do processo consiste na avaliação das métricas de dependabilidade (por exemplo, disponibilidade e confiabilidade). Por fim, os resultados são obtidos.

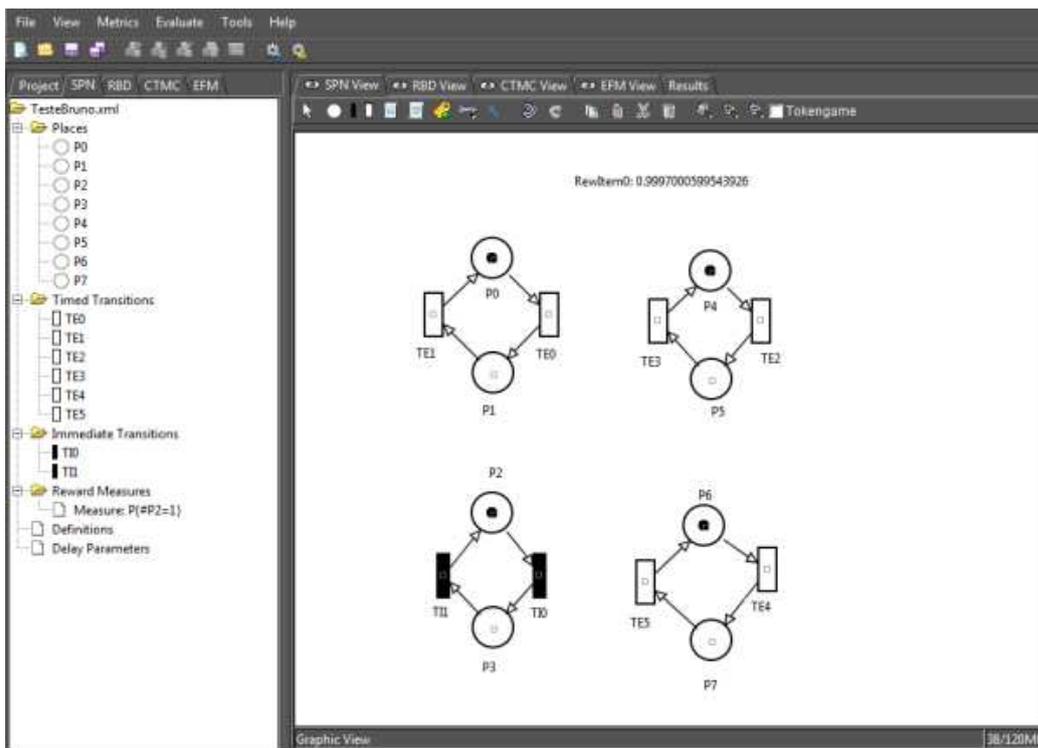


Figura 1.2: Exemplo de modelo SPN na ferramenta Mercury

Caso se escolha a avaliação do modelo de fluxo de energia, deve-se possuir a arquitetura elétrica em alto nível do ambiente que se deseja modelar. É importante ressaltar que subsistemas podem ser gerados para atenuar a complexidade do sistema final.

Em seguida é feita a representação da arquitetura de alto nível no modelo EFM. Neste

ponto tem-se dois caminhos a seguir, que são: a aplicação ou não do algoritmo ADCE. Caso não seja de interesse do projetista, o próximo passo é a avaliação do modelo EFM sem a otimização do algoritmo proposto e posteriormente a obtenção dos resultados dessa avaliação. Por outro lado, se for escolhida a opção de otimização do ADCE, os pesos das arestas que direcionam e quantificam o fluxo serão otimizados. Este procedimento é feito considerando as eficiências energéticas de cada componente e respeitando as restrições de capacidade de energia que cada dispositivo é capaz de fornecer (sistema elétrico) ou extrair (sistema de refrigeração).

Após a otimização do fluxo elétrico, as seguintes métricas podem (dependendo de cada modelo) ser melhoradas: exergia operacional (utilizada para medir o impacto ambiental do sistema em relação ao consumo elétrico), custo operacional (representa o custo de operação do sistema em determinado período de tempo), energia necessária ao funcionamento da arquitetura e eficiência do sistema. É importante destacar que em nenhum dos casos estudados o resultado otimizado foi inferior ao resultado sem otimização. No pior dos casos, os resultados obtidos foram iguais àqueles da avaliação sem a utilização do algoritmo. Este fato foi observado quando não há caminhos alternativos ao fluxo elétrico na arquitetura.

Após esta etapa, é possível avaliar o modelo EFM de forma idêntica a não aplicação do ADCE, uma vez que a otimização do fluxo não altera as propriedades do modelo. Com a finalidade de demonstrar a aplicabilidade do algoritmo ADCE, os resultados obtidos considerando sua execução, podem ser comparados com os resultados dos modelos que não foram otimizados.

1.5 Trabalhos Relacionados

Nos últimos anos, algumas pesquisas foram desenvolvidas sobre a tentativa de redução do consumo de energia elétrica em centros de dados e nuvens. Um subconjunto destas pesquisas também considerou o impacto sobre as questões de sustentabilidade e custo. Em outro subconjunto foi proposta a utilização de técnicas de otimização para gestão da energia elétrica.

Gandhi [32] realiza uma análise da eficácia do gerenciamento dinâmico de energia em centros de dados. Esta política reduz o consumo de energia do ambiente avaliado, para isto, desligam-se os servidores quando os mesmos se encontram ociosos. É feita uma comparação sobre o tempo de preparação de um servidor que estava desligado até

que o mesmo torne-se disponível e o impacto deste tempo no desempenho do sistema. No entanto, a natureza altamente abstrata do trabalho, não considera os componentes elétricos do centro de dados, o que pode tornar os resultados irreais.

Uma abordagem interessante é apresentada por Chen [19] com uma solução integrada do gerenciamento dos componentes de TI e da gestão da infraestrutura de refrigeração, para melhorar a eficiência global das operações do centro de dados. Porém, neste trabalho os autores não propuseram uma estratégia para otimizar o fluxo elétrico. Nesta dissertação isto é proposto.

Em Gyarmati [41] os autores discutem o impacto de parâmetros arquitetônicos no consumo de energia do centro de dados. O consumo de energia das seguintes estruturas de centro de dados é abordado: BCube [40], DCell [39], e Fat-tree [38]. Os resultados desse estudo demonstram que o consumo de energia dos centros de dados são maiores que o necessário, uma vez que os projetos são concebidos para um número muito maior de servidores que aqueles que de fato são necessários. Nesta dissertação é possível estimar a energia necessária para cada tamanho de centro de dados, uma vez que utiliza uma ferramenta de simulação para modelos, chamada Mercury.

Yamini [64] propôs um algoritmo para relacionar a utilização de recursos do processador com o consumo de energia. O algoritmo usa o método de *Pareto* e distância *euclidiana* para maximizar a utilização dos recursos do processador a fim de reduzir o consumo de energia. Funções de custo são incorporadas a esta heurística para encontrar possibilidades de economia de energia. Embora este trabalho utilize uma heurística para o consumo de energia, os autores mantêm o foco nos servidores (ou computadores) da infraestrutura de TI, a medita que esta dissertação considera o consumo de infraestrutura elétrica como foco principal.

Mukherjee [44] propôs algoritmos para reescalonar serviços de CPUs sobrecarregadas, ao passo que identificam CPUs ociosas em uma nuvem privada. Assim, propõe uma computação em nuvem "ecologicamente amigável" (*eco-friendly*), com foco na infraestrutura de TI da nuvem. Este trabalho considerou uma nuvem configurada com a plataforma de infraestrutura como serviço (IaaS). Dois algoritmos foram utilizados, o de "colônia de abelhas" para o reescalonamento de serviços de CPUs sobrecarregadas e o de "colônia de formigas" para encontrar CPUs ociosas, considerando a gestão do consumo de energia. Esses algoritmos são conhecidos da área de otimização e embora sejam muito interessantes, neste trabalho os autores focam na redução do consumo de energia da nuvem de acordo com o escalonamento de carga dos servidores, enquanto uma aplicação

voltada a infraestrutura elétrica certamente seria mais eficiente, como demonstrado nesta dissertação.

Baliga [9] analisa o consumo de energia em nuvens públicas e privadas, tendo em vista a comunicação de rede, processamento e armazenamento de dados. A análise demonstra que há um equilíbrio entre o consumo de energia dos dispositivos da nuvem. Esta análise, aborda apenas a rede de fluxo de dados dentro da nuvem, e a economia de energia que determinado fluxo pode causar sobre outro. A redução apresentada é muito inferior ao da alternativa de fluxo na infraestrutura elétrica.

Xu [63] apresenta um modelo para o gerenciamento de energia em uma nuvem privada. O modelo considera o consumo de energia e eficiência dos servidores para dissociar a aplicação da infraestrutura física. Este trabalho considera a eficiência energética dos servidores da nuvem, mas se limita a gestão de energia dos mesmos, não considerando a infraestrutura elétrica.

Curtis propõe uma ferramenta chamada "REWIRE"[23], um framework baseado em otimização para design de centros de dados. Com o REWIRE é possível maximizar a eficiência das arquiteturas de rede dos centros de dados por intermédio de um algoritmo de otimização. Ademais, a ferramenta utiliza um algoritmo que minimiza o *delay* ponta a ponta das transmissões de dados e melhora a largura de banda de rede. Embora a ferramenta forneça um apoio ao projetista do centro de dados, suas funcionalidades se limitam ao gerenciamento da rede de dados entre os servidores, não tratando a infraestrutura elétrica, assim como o Mercury.

Chao propôs um sistema para redirecionar o fluxo de dados sem interrupção pelos IPs e roteadores em uma rede [61]. O sistema se chama PROBE ("sonda"), e é mais uma solução para os problemas de fluxo de rede. Mesmo o sistema sendo aplicado a resolução de um problema de fluxo, seu foco está na rede de dados, enquanto essa dissertação se preocupa com o fluxo da infraestrutura elétrica em centros de dados ou nuvens.

A pesquisa desses trabalhos foi feita no âmbito de: otimização, problemas de fluxo, redução do consumo elétrico e questões de sustentabilidade, tanto para centros de dados quanto para nuvens. Os trabalhos encontrados retratam uma preocupação em geral com a diminuição do consumo de energia elétrica em centros de dados ou nuvens. Porém, geralmente, se aplicam a infraestrutura de TI dos ambientes ou no fluxo de dados dos mesmos (inclusive as ferramentas) e não na infraestrutura elétrica, como tratado de forma pioneira pelo grupo de pesquisas MoDCS do Centro de Informática da UFPE e estendido por essa dissertação.

1.6 Organização da Dissertação

No capítulo introdutório foram apresentados os objetivos, a motivação, a justificativa, método proposto e os trabalhos relacionados da dissertação. Ademais, o principal assunto do trabalho foi contextualizado. O restante da dissertação está organizado da seguinte maneira:

O Capítulo 2 apresenta os principais aspectos teóricos, base deste trabalho, em relação à infraestrutura de centro de dados, computação em nuvem, dependabilidade, diagrama de blocos de confiabilidade (RBD), sustentabilidade, modelagem de fluxo energético (EFM), grafos acíclicos dirigidos e custos. O Capítulo 3 apresenta a principal contribuição da presente dissertação, detalhando o algoritmo de distribuição de carga elétrica (ADCE). Na sequência, o Capítulo 4 é composto pela apresentação de dois estudos de caso, que representam arquiteturas elétricas reais de nuvens e centros de dados, com o objetivo de validar o algoritmo proposto, ilustrando as possíveis melhorias. Por fim, o Capítulo 5 apresenta as considerações finais para este trabalho, aponta as contribuições alcançadas, e sugere alguns trabalhos futuros.

2

Fundamentação Teórica

Este capítulo apresenta os fundamentos teóricos que alicerçaram o desenvolvimento da presente dissertação. Inicialmente, serão apresentados os conceitos dos centros de dados e suas subdivisões. Em seguida, será abordado o paradigma da computação em nuvem, juntamente com uma explicação acerca do *software* utilizado nos experimentos mencionados nesta dissertação. Algumas definições de sustentabilidade serão apresentadas ao longo do trabalho, bem como um resumo sobre as questões de dependabilidade. Para finalizar é proporcionada uma introdução aos modelos RBD e EFM.

Tais definições são fundamentais para que seja proporcionado o conhecimento necessário ao entendimento do algoritmo proposto para otimização do fluxo elétrico nos modelos EFM expostos no Capítulo 3 e dos estudos de caso constantes no Capítulo 4.

2.1 Infraestrutura de Centro de Dados

Os centros de dados surgiram juntamente com a indústria da computação, quando os primeiros sistemas computacionais exigiam ambientes em condições especiais para funcionamento e pela complexidade em sua operação e manutenção [7]. Com o passar das décadas, ocorreram muitas alterações, como por exemplo, a organização, armazenamento e processamento dos centros de dados. Na literatura existem várias definições sobre os centros de dados. A definição proposta neste trabalho foi a de Marin [47] onde afirma que em um centro de dados *"possuem equipamentos responsáveis pelo processamento e armazenamento de informações cruciais para a continuidade de negócios nos mais variados tipos de organizações, sejam elas empresas, instituições de ensino, indústrias, órgãos governamentais, hospitais, hotéis, entre outros"*.

É perceptível o crescimento dos centros de dados, quer com a criação de novas instalações, quer com a expansão das existentes. Há grandes organizações operando vários centros de dados em todo o mundo, cada um com dezenas ou centenas de servidores. Esses centros de dados facilitam bastante a vida do homem moderno, possibilitando, inclusive, armazenamento de dados, computação, melhoria na comunicação, além de possibilitar a colaboração e permitir o entretenimento. É por essa razão que, os centros de dados devem ser robustos e flexíveis para suportar novos requisitos tecnológicos.

Os centros de dados podem ser classificados de acordo com seus níveis de redundância de equipamentos, fluxo elétrico e de refrigeração. Turner, Seader e Brill [58] afirmam que o projeto global de um centro de dados pode ser classificado em quatro níveis, a saber:

- **Projeto de centro de dados nível I**

Possui um caminho único para refrigeração e distribuição de energia, sem componentes redundantes.

- **Projeto de centro de dados nível II**

Contém componentes redundantes, melhorando a disponibilidade.

- **Projeto de centro de dados nível III**

Possui múltiplos caminhos de distribuição de eletricidade para o resfriamento, porém apenas um caminho ativo. Fornecem redundância mesmo durante a manutenção.

- **Projeto de centro de dados nível IV**

Mantém múltiplos caminhos de distribuição de eletricidade para o resfriamento, além de dois caminhos ativos. É dotado de componentes redundantes em cada caminho e suporta qualquer falha de um único equipamento sem impactar em seu funcionamento.

Além dos níveis, podemos classificar os centros de dados quanto aos seus tipos, em: *enterprise*, *internet* e *collocation*. Embora Marin esclareça que um centro de dados é formado por vários sistemas [47], como:

- Sala de Computadores (computing room);
- Ar condicionado e controle ambiental;
- Distribuição elétrica e UPS (Uninterruptible Power Supply);

- Automação do edifício;
- Detecção e supressão de incêndio;
- Segurança e controle;
- Espaço de suporte, entre outros.

Neste trabalho consideramos a proposta de Arregoces e Portolani [6], onde um sistema de centro de dados genérico é adotado, para fins de representação e modelagem, consistindo, dentro desse contexto, essencialmente, nos seguintes subsistemas: (i) infraestruturas de TI, (ii) infraestrutura de resfriamento, e (iii) infraestrutura elétrica. A Figura 2.1 exemplifica esta proposta e direciona o fluxo da energia.

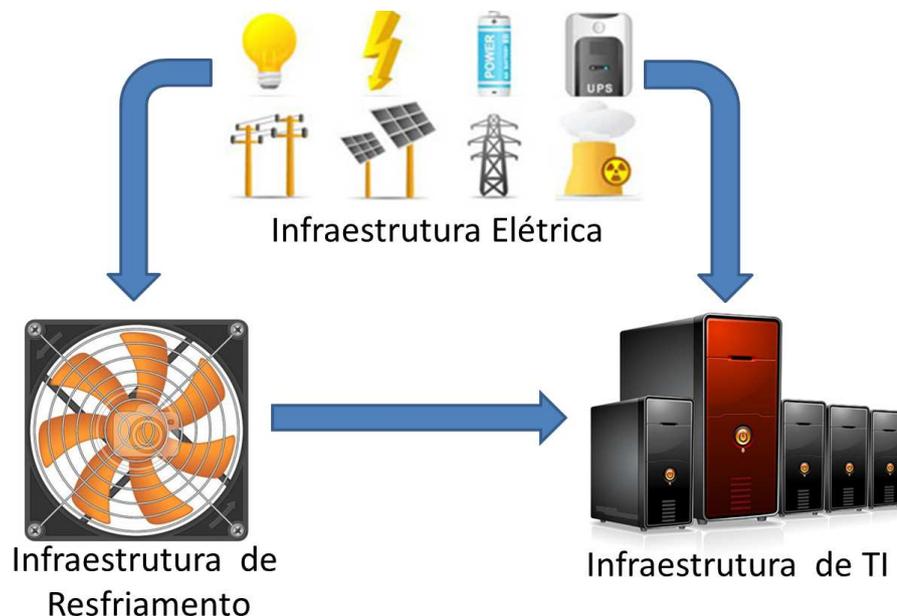


Figura 2.1: Subsistemas de centro de dados

A seguir, demonstramos um detalhamento sobre estas infraestruturas:

- **(i) Infraestrutura de TI**

Consiste em três componentes principais: servidores, equipamentos de rede e dispositivos de armazenamento. Os serviços de software são normalmente organizados em uma arquitetura multicamadas separadas em servidores web, de aplicações e de banco de dados. A infraestrutura de rede inclui o hardware físico adotado para transmitir dados eletronicamente, tais como switches, hubs, roteadores, pontes e

gateways. Os serviços de software são normalmente organizados em uma arquitetura multicamadas, separadas em servidores web, de aplicações e de banco de dados [10].

- **(ii) Infraestrutura de Resfriamento**

É responsável por controlar a temperatura ambiente, removendo o excesso de calor gerado pela dissipação de energia das infraestruturas de TI e elétrica, resfriando, conseqüentemente, o ambiente. A temperatura é regulada no sistema de resfriamento, a fim de mantê-la entre 20 e 22 graus Celsius [52]. A infraestrutura de refrigeração da sala de computação é composta basicamente por ar-condicionado (CRAC), chillers e torres de resfriamento.

- **(iii) Infraestrutura Elétrica**

É responsável pelo fornecimento ininterrupto de energia, condicionada à tensão e à frequência corretas, para os equipamentos de TI hospedados em racks de centros de dados, bem como à infraestrutura de refrigeração [30]. A energia provida pelo sistema público de distribuição passa, normalmente, pelos transformadores, chaves de transferência, fontes de alimentação ininterruptas (UPS), PDUs e, por fim, pelas régua de energia do rack [14] [45].

2.2 Computação em Nuvem

A computação em nuvem (*cloud computing*), pode ser intitulada como o estágio atual da evolução da internet, onde são fornecidos vários tipos de serviços, como infraestrutura de computação, de aplicativos e de processos de negócios para a colaboração pessoal, tudo isso de acordo com as necessidades do usuário.

O uso da computação em nuvem integra a rotina de muitos usuários, embora seu conceito seja transparente e muitos destes não percebiam que a estão utilizando, como no caso do uso de sites de busca, redes sociais e microblogs. A Figura 2.2, demonstra a utilização das nuvens por diversos meios, como notebook, tablet, computador pessoal, celular e variadas aplicações, como gmail, facebook, youtube, twitter e google docs, bem como exemplifica seu acesso distribuído e remoto.

Os tipos de serviços oferecidos na computação em nuvem são variados. As plataformas de *hardware* e *software* são fornecidos em forma de serviços sob demanda. Os mais comuns são: Plataforma como um Serviço (PaaS), Software como um Serviço (SaaS) e

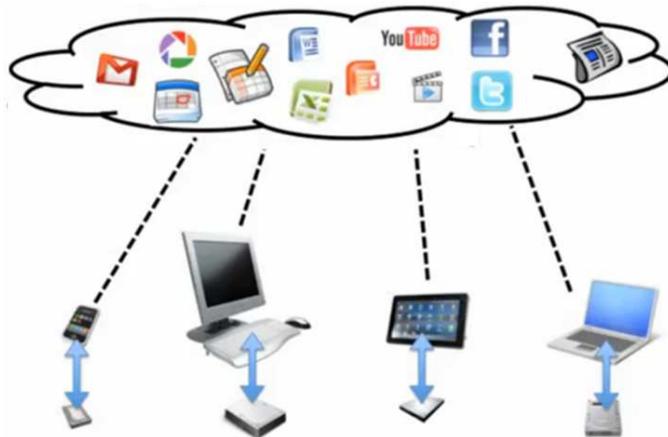


Figura 2.2: Exemplo de utilização da computação em nuvem

Infraestrutura como um Serviço (IaaS) [29] [36].

Quando se analisa os serviços oferecidos por provedores de nuvens, muitas questões devem ser consideradas. Mas a necessidade do cliente certamente é a principal. Se o cliente deseja reduzir custos, aumentar a confiabilidade ou obter alto desempenho, deverá, ele, procurar uma nuvem que supra suas necessidades. As nuvens são divididas de acordo com a natureza e o controle de acesso aos seus recursos físicos e virtuais. Assim, existem três tipos principais de nuvens, ou seja, diferentes formas de acessar dados através da web, quais sejam: pública, privada e híbrida.

As nuvens são configuradas com sistemas específicos, de acordo com sua natureza. O Eucalyptus (*Elastic Utility Computing Architecture Linking Your Programs To Useful Systems*) é um software que implementa nuvens privadas e híbridas. Foi criado com a finalidade de servir de ambiente para pesquisas e sua interface é compatível com os serviços comerciais da Amazon EC2 [28].

A nuvem utilizada nesta dissertação foi configurada com o sistema *Eucalyptus*, é classificada como do tipo privada e utiliza o modelo de negócio de Infraestrutura como um Serviço (IaaS).

2.3 Dependabilidade

Com o crescimento das tecnologias utilizadas através da Internet, a confiabilidade tornou-se um atributo de grande relevância no desenvolvimento, implantação e operação de hardware ou software utilizado nessa grande rede [46]. A replicação de serviços é

normalmente disponibilizada através de hosts distribuídos mundialmente, de modo que, na ocorrência de falha em algum serviço, sua replicação esteja pronta para assumir a demanda [37]. Um serviço é confiável quando implementa a funcionalidade especificada pelo sistema. A falha do sistema ocorre quando não consegue fornecer sua funcionalidade especificada. De fato, o conceito de dependabilidade também se associa à confiabilidade e à tolerância a falhas.

Na literatura é possível encontrar várias definições para dependabilidade, uma delas a define como "*a habilidade de disponibilizar um conjunto de serviços com um determinado nível de confiança*" [27], já outra como "*o sistema que executará ações específicas ou apresentar resultados específicos de maneira confiável e em tempo hábil*" [51].

A confiabilidade de um item ou sistema "*é a probabilidade que um sistema desempenha satisfatoriamente (ou adequadamente) o seu propósito específico por um dado período até a ocorrência da primeira falha*" [46]. Como exemplo, um sistema com confiabilidade de 98% para 8760 horas, significa que há 98% de chances de o sistema não falhar de 0 até 8760 horas de operação.

A disponibilidade é também outro conceito importante, que "*quantifica o efeito misto dos processos de falhas e reparo em um sistema*". Em geral, a disponibilidade e a confiabilidade são conceitos relacionados, mas diferem no sentido, uma vez que o primeiro pode considerar a manutenção de componentes que falharam [27].

Em Avizienis [8], é apresentada uma exposição sistemática dos conceitos de dependabilidade: atributos, meios e ameaças. A Figura 2.3 exibe essa classificação, onde:

- **Os atributos:** possibilitam a obtenção de medidas quantitativas para a análise dos serviços oferecidos;
- **Os meios:** são as formas pelas quais a dependabilidade é atingida;
- **As ameaças:** compreendem as falhas, erros e defeitos. A falha do sistema é o evento que ocorre quando a entrega do serviço não acontece como desejado.

As métricas de dependabilidade (disponibilidade e confiabilidade) podem ser computadas através de modelos combinatórios, como o RBD, ou de modelos baseados em estados, como o SPN, entre outras formas. Os RBDs permitem estabelecer equações de forma fechada e representar redes de componentes. O objeto da análise sobre as questões de dependabilidade está nos atributos que a compreende: disponibilidade, confiabilidade,

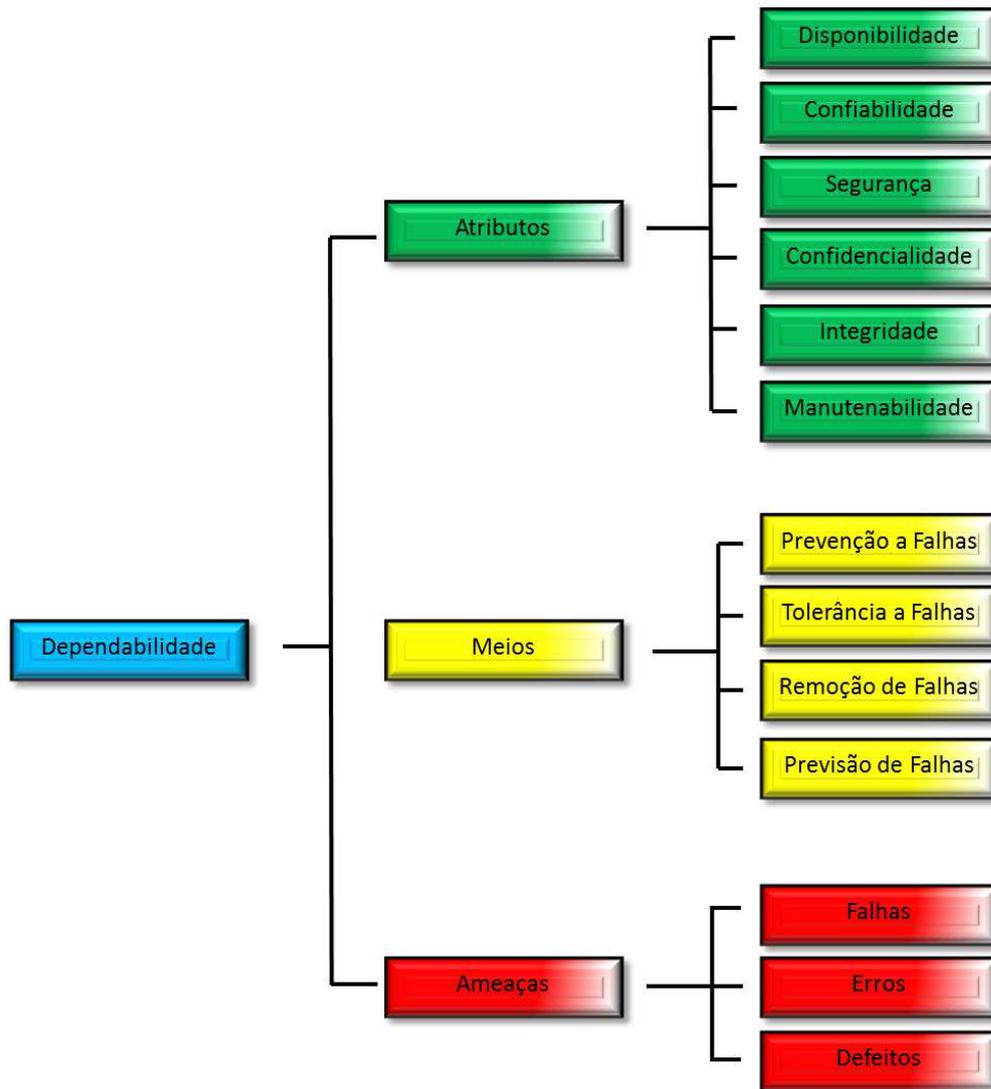


Figura 2.3: Árvore de Dependabilidade Adaptada de Avizienis [8]

segurança, confidencialidade, integridade e manutenibilidade [8], como exibidos na árvore de dependabilidade (Figura 2.3). Porém, nesse trabalho apenas a disponibilidade e a confiabilidade serão explicadas em detalhes. Para conhecimento sobre as demais métricas, o leitor é redirecionado a [46].

Para cálculo das métricas de dependabilidade, é necessário conhecer os tempos de falha (Time to Fail - TTF) e o tempo do reparo (Time to Repair - TTR) dos componentes presentes nos modelos. Entretanto, como o tempo de atividade e de inatividade, geralmente não são conhecidos, a opção mais plausível é da utilização de uma média sobre essas métricas. Assim, é considerado um tempo médio de falha (Mean Time to Fail -

MTTF) e um tempo médio do reparo (Mean Time to Repair - MTTR).

A disponibilidade (A) de estado estacionário pode ser calculada pela equação 2.1:

$$A = \frac{MTTF}{MTTF + MTTR} \quad (2.1)$$

Considerando um determinado período de tempo representado pelo intervalo $(0, t)$, $R(t)$ representa a probabilidade do componente funcionar (não falhar) desde 0 até t . Ao considerar o tempo de falha de uma distribuição exponencial, a confiabilidade é representada por:

$$R(t) = \exp \left[- \int_0^t \lambda(t') dt' \right] \quad (2.2)$$

onde $\lambda(t')$ é a taxa de falha instantânea.

Através de análise transiente ou de simulação, a confiabilidade (R) é obtida, e, em seguida, o $MTTF$ pode ser calculado, bem como o desvio padrão do tempo até a falha (TTF):

$$MTTF = \int_0^{\infty} t f(t) dt = \int_0^{\infty} - \frac{dR(t)}{dt} t dt = \int_0^{\infty} R(t) dt \quad (2.3)$$

$$sd(TTF) = \sqrt{\int_0^{\infty} t^2 f(t) dt - (MTTF)^2} \quad (2.4)$$

Deve-se salientar que o RBD pode ser avaliado através de simulação, sempre que adotadas distribuições mais complexas que a exponencial.

Deve-se ter em mente que, para o cálculo da confiabilidade de um determinado serviço do sistema, a atividade de reparação do respectivo serviço não deve ser representado. Além disso, considerando a Equação 2.1 e que a $UA = 1 - A$ (indisponibilidade), a seguinte equação pode ser derivada:

$$MTTR = MTTF \times \frac{UA}{A} \quad (2.5)$$

A derivação padrão do tempo de reparo (TTR) pode ser calculada da seguinte forma:

$$sd(TTR) = sd(TTF) \times \frac{UA}{A} \quad (2.6)$$

Em seguida, $\frac{MTTF}{sd(TTF)}$ (e $\frac{MTTR}{sd(TTR)}$) podem ser computadas para a escolha da distribuição polinomial que melhor se adapta as distribuições TTF e TTR [24] [57].

2.3.1 Medida de importância para Componentes

A importância dos componentes é um parâmetro que indica o impacto que um componente em particular exerce na confiabilidade global do sistema. Na presente dissertação, a métrica de importância da confiabilidade (*reliability importance* - RI) e a importância da confiabilidade e do Custo (*reliability and cost importance* - RCI) são utilizadas para identificar os componentes mais críticos à confiabilidade do sistema. As linhas a seguir detalham cada um desses índices.

A **importância para confiabilidade (RI)** é uma métrica que pode ser útil para a identificação dos componentes mais críticos de um sistema. *RI* (ou *Birnbaum Importance*) de um componente i corresponde à quantidade de melhoria na confiabilidade do sistema, quando a confiabilidade do componente i é aumentada em uma unidade [45]. O RI do componente i pode ser calculado como:

$$RI_i = R_s(1_i, \mathbf{p}^i) - R_s(0_i, \mathbf{p}^i) \quad (2.7)$$

Em que RI_i é a importância da confiabilidade do componente i ; \mathbf{p}^i representa o vetor de confiabilidade dos componentes com o componente i removido; 0_i representa a falha do componente i , e 1_i denota o componente i funcionando corretamente.

De acordo com a definição anterior, o valor do *RI* representa um indicador que aponta o principal componente para se melhorar a confiabilidade do sistema. RI_i depende somente da estrutura do sistema e da confiabilidade dos demais componentes. Ou seja, a *RI* do componente i é determinada pelas confiabilidades de todos os componentes, exceto i [53] [45].

Kuo e Zou [45] afirmam que em uma estrutura em série, o componente menos confiável possui o maior valor de *RI*. Já em estruturas formadas por componentes em paralelo, o componente mais confiável terá o maior valor de *RI*. Baseado nessas conjecturas, o componente com maior valor de *RI* deveria ser o alvo inicial para se

melhorar a confiabilidade do sistema.

Entretanto, essa abordagem nem sempre pode ser aplicada na prática, devido aos custos dos equipamentos [45]. Por exemplo, se o maior RI de um sistema aponta para o componente de maior valor financeiro do ambiente, é possível que, melhorados os componentes de menor valor financeiro (mesmo os de menor valor do RI), a confiabilidade do sistema aumente significativamente. Para solucionar esse problema, o ideal seria utilizar uma métrica que relacione a confiabilidade com os custos dos equipamentos.

A **Importância para Confiabilidade e Custo** (RCI) é uma métrica que relaciona o RI previamente abordado, com os custos de aquisição dos componentes do sistema. A RCI do componente i pode ser calculada como:

$$RCI_i = RI_i \times \left(1 - \frac{C_i}{C_{sys}}\right) \quad (2.8)$$

Em que RCI_i é a importância para confiabilidade e custo propostos; RI_i é a importância para confiabilidade do componente i ; C_i é o custo de aquisição do componente i e C_{sys} é o custo de aquisição do sistema.

Esta métrica foi utilizada para identificar a ordem de redundância dos componentes presentes nas arquiteturas de base dos estudos de caso, tanto para a infraestrutura elétrica da nuvem, quanto do centro de dados.

2.4 Modelo de Dependabilidade

Diagrama de Bloco de Confiabilidade

O diagrama de bloco de confiabilidade (*Reliability Block Diagram - RBD*) é um modelo combinatório que foi inicialmente proposto como uma técnica para calcular a confiabilidade de sistemas através de diagramas de blocos intuitivos. Em geral, os diagramas de blocos oferecem uma representação gráfica dos conectores e dispositivos do sistema, que, uma vez conhecido o estado de seus componentes, podem ser utilizados para determinar o estado geral do sistema [46].

A estrutura do RBD estabelece a interação lógica entre os componentes, definindo quais combinações de elementos ativos são capazes de conservar o funcionamento do sistema. Assim, o sistema é representado pelos subsistemas ou componentes relacionados

2.4. MODELO DE DEPENDABILIDADE DIAGRAMA DE BLOCO DE CONFIABILIDADE

com sua função ou da relação de confiabilidade [62].

Em RBD, é possível representar um componente físico no modo operacional por um bloco. Logo, para representar uma falha de um componente, é necessário remover o bloco correspondente ao componente do modelo. Se houver pelo menos um caminho que faça a ligação entre os pontos de entrada e de saída, o sistema continua funcionando corretamente. Em outras palavras, se removidas uma quantidade suficiente de blocos para interromper a conexão entre os pontos de entrada e de saída, o sistema falha [26] [45]. Essa técnica também foi estendida para calcular outras métricas de dependabilidade, tais como a disponibilidade e a manutenibilidade.

RBDs foram adotados para avaliar estruturas em série-paralelo e estruturas mais genéricas, como: pontes, estrelas e disposições em delta. Os RBDs mais simples e mais comuns suportam apenas estruturas série-paralelo [46], estas estruturas serão abordadas em detalhes a seguir.

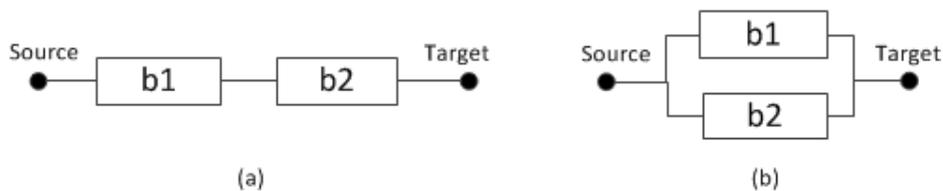


Figura 2.4: (a) Blocos em Série; (b) Blocos em Paralelo

A Figura 2.4 ilustra dois exemplos, em que os blocos são organizados em série (Figura 2.4(a)) e em paralelo (Figura 2.4(b)). Com a disposição em série, se há o fracasso de um único componente, todo o sistema falha [17]. Considerando um sistema com n elementos em série, a confiabilidade é obtida através da equação:

$$R_s(t) = \prod_{i=1}^n R_i(t) \quad (2.9)$$

em que $R_s(t)$ é a confiabilidade do sistema em série no tempo t e $R_i(t)$ é a confiabilidade do bloco b_i .

Já para a disposição em paralelo (ver Figura 2.4(b)), é necessário que pelo menos um componente esteja funcionando para que todo sistema esteja operacional. Considerando um sistema com n elementos em paralelo, a confiabilidade é obtida através da equação:

$$R_p = 1 - \prod_{i=1}^n (1 - R_i) \quad (2.10)$$

Em que $R_p(t)$ é a confiabilidade do sistema em paralelo no tempo t e $R_i(t)$ é a confiabilidade do bloco b_i .

Para que o leitor tenha conhecimento de outros exemplos e equações de forma fechada, é recomendado que se dirija a [45] [46].

2.4.1 Função Estrutural

Funções estruturais RBD são funções matemáticas discretas usadas para representar relações entre componentes individuais de um sistema e o sistema como um todo.

Considere um sistema S composto por um conjunto de componentes, $C = c_i \mid 1 \leq i \leq n$, onde o estado do sistema S e os componentes podem estar operacionais ou falhos e n é o número de componentes do sistema. Seja X_i uma variável aleatória discreta, indicando o estado do componente i , assim:

$$X_i = \begin{cases} 0 & \text{Se o componente falha,} \\ 1 & \text{Se o componente está operacional;} \end{cases}$$

Além disso, cada componente do vetor $X = (X_1, X_2, \dots, X_n)$ representa o estado de cada componente do sistema. O estado do sistema é determinado pelos estados dos componentes. A função de estrutura $\phi(X)$ mapeia o vetor X para 1 ou 0, como mostrado abaixo:

$$\phi(x) = \begin{cases} 0 & \text{Se o sistema falha,} \\ 1 & \text{Se o sistema está operacional;} \end{cases}$$

Ora, se o estado de todos os componentes do sistemas é conhecido, então o estado do sistema também o é. Logo, o estado do sistema é uma função determinística do estado de todos os componentes. Veja a Equação 2.11 a seguir:

$$\phi = \phi(X) = \phi(X_1, X_2, \dots, X_n) \quad (2.11)$$

onde $\phi(X)$ é a função estrutural do sistema.

As regras de formação das funções estruturais, para componentes em série e paralelo são detalhadas a seguir:

Componentes em Série:

Sejam n componentes x_1, x_2, \dots, x_n em série, a função estrutural ϕ desses componentes é representada pela Equação 2.12 a seguir:

$$\phi(x) = 1 - \prod_{i=1}^n (x_i) = \min(x_1, x_2, \dots, x_n) \quad (2.12)$$

Sejam n componentes x_1, x_2, \dots, x_n em paralelo, a função estrutural ϕ desses componentes é representada pela Equação 2.13 a seguir:

$$\phi(x) = 1 - \prod_{i=1}^n (1 - x_i) = \max(x_1, x_2, \dots, x_n) \quad (2.13)$$

2.4.2 Função Lógica

Com o mesmo objetivo das funções estruturais, as funções lógicas indicam um relacionamento entre os estados dos componentes e o estado do sistema. Entretanto, em alguns casos não é fácil simplificar funções estruturais para uma forma minimizada [45]. Técnicas de minimização booleanas podem ser utilizadas em funções lógicas para simplificar ou minimizar as funções geradas. Em seguida são apresentadas as equações para calcular as funções lógicas de disposições em série e paralelo.

Componentes em Serie:

Seja $x = (x_1, x_2, \dots, x_n)$ o vetor que representa o estado de n componentes de um sistema. A função lógica (S) que representa o funcionamento do sistema em série é:

$$serial(x) = (x_1 \wedge x_2 \wedge \dots \wedge x_n) \quad (2.14)$$

Componentes em Paralelo:

Seja $x = (x_1, x_2, \dots, x_n)$ o vetor que representa o estado de n componentes de um sistema. A função lógica (S) que representa o funcionamento do sistema em paralelo é:

$$Paralelo(x) = (x_1 \vee x_2 \vee \dots \vee x_n) \quad (2.15)$$

Para mais detalhes sobre estas funções lógicas e estruturais o leitor é encaminhado a [45] [46]. No capítulo 4 desta dissertação é possível acompanhar um exemplo de geração de funções lógicas e estruturais em uma modelagem de RBD feita para a infraestrutura elétrica de um centro de dados.

2.5 Sustentabilidade

O termo sustentabilidade tem sido utilizado desde a década de 1980, com o sentido da sustentabilidade humana no planeta Terra. Isso resultou na definição mais utilizada nos dias de hoje, como parte do conceito de desenvolvimento sustentável: *"O desenvolvimento sustentável é o desenvolvimento que satisfaz as necessidades do presente sem comprometer a capacidade das gerações futuras de satisfazerem as suas próprias necessidades"* [13].

Nas últimas décadas, com o aumento da preocupação sobre as questões de sustentabilidade em todo o planeta, surgiu o conceito de Computação Verde (*Green Computing*), que é definida como: *"o estudo e a prática da computação ecologicamente sustentável, incluindo concepção, fabricação, uso e descarte de computadores, servidores e subsistemas associados (monitores, impressoras, dispositivos de armazenamento e sistemas de comunicação) com eficiência e eficácia. Dessa maneira, causando o menor impacto possível ao meio ambiente [50]"*. Muitas pesquisas publicadas nos últimos anos abordam os centros de dados verdes (*green data center*), mais especificamente foram estudados os trabalhos publicados com foco na redução do consumo de energia: [15] [65] [18] [35] [14], o que é altamente relevante para as empresas de TI e estão em sintonia com este trabalho.

De acordo com Bouley [12], para se medir o nível de sustentabilidade de um centro de dados, deve-se primeiramente determinar seu consumo de energia e, em seguida, utilizar uma métrica para fazer a avaliação da sustentabilidade. Na verdade, além do consumo de energia, existem outros métodos interligados de comparação de equipamentos a partir do ponto de vista da sustentabilidade. Entre outros, como exemplo, podem ser citados os materiais utilizados na fabricação dos equipamentos e os prejuízos ambientais irreversíveis para as futuras gerações.

O impacto ambiental pode ser quantificado também pela métrica utilizada na segunda lei da termodinâmica, exergia [25]. A exergia é uma métrica que estima a eficiência de um sistema, por exemplo, baseado no consumo elétrico de uma arquitetura de um centro

de dados. Como um índice para a avaliação global da sustentabilidade, a propriedade termodinâmica da exergia é de grande importância.

Segundo Dincer [25] a exergia é definida como "*a fração máxima de energia latente que pode ser teoricamente convertida em trabalho útil*". Uma ilustração interessante seria comparar a exergia de um kJ de gasolina com a de um kJ de água à temperatura ambiente. A exergia da gasolina é muito maior, uma vez que pode ser usada para movimentar um caminhão, mas a água à temperatura ambiente não. A exergia é calculada como o produto da energia por um índice de fator de qualidade, como mostrado na Equação 2.16.

$$Exergy = Energy \times \alpha \quad (2.16)$$

O índice α é um fator de qualidade representado pela relação da exergia/energia, conforme apresentado na Tabela 2.1 [25].

Tabela 2.1: Valores de energia e exergia para diferentes modalidades energéticas.

Origem	Energia (J)	Exergia (J)	α
Água a 80 graus Celsius	100	16	0.16
Vapor a 120 graus Celsius	100	24	0.24
Gás Natural	100	99	0.99
Eletricidade	100	100	1.00

Na presente dissertação foi utilizada a exergia operacional para medir o impacto ambiental de desperdício de energia dos componentes elétricos do centro de dados. Cada componente elétrico dissipa uma fração do calor, que não é teoricamente convertida em trabalho útil, denominada então de exergia operacional. A equação 2.17 é usada para computar o valor da exergia operacional.

$$Ex_{op} = \sum_{i=1}^n Ex_{opi} \times T \times (A + \alpha(1 - A)) \quad (2.17)$$

onde Ex_{opi} é a exergia operacional de cada dispositivo; T é o período da análise; A é a disponibilidade do sistema e α é o fator que representa a quantidade de energia que continua a ser consumida após a falha de determinado dispositivo.

Cada tipo de dispositivo usa uma equação específica para calcular sua exergia operacional. Assim, a equação usada para avaliar um gerador a diesel é diferente da usada para avaliar uma torre de resfriamento. Nessa dissertação apenas equipamentos elétricos foram considerados. A Equação 2.18 é utilizada para computar a exergia de cada dispositivo

elétrico.

$$Exergy = P_{in} \times (1 - \eta) \quad (2.18)$$

onde P_{in} é a energia elétrica de entrada de cada equipamento e η é a eficiência informada;

Para maiores informações, o leitor é redirecionado a [18].

2.6 Modelagem de Fluxo Energético

O modelo de fluxo elétrico (*Energy Flow Model* - EFM [16]) quantifica o fluxo de energia entre os componentes de um sistema, respeitando os limites de energia que cada componente pode fornecer (considerando os dispositivos elétricos) ou a máxima capacidade de resfriamento (assumindo os dispositivos de refrigeração). Nessa dissertação, apenas a infraestrutura elétrica dos ambientes foi considerada. Ademais, é importante salientar que as unidades de medida utilizadas pela modelagem EFM para mensurar o valor da energia é o kJ para modelos de refrigeração e kW para modelos elétricos. O EFM é representado por um grafo acíclico dirigido, no qual os componentes são modelados como vértices e as respectivas ligações correspondem às arestas.

A seguir, a definição formal do EFM:

$G = (N, A, w, f_d, f_c, f_p, f_\eta)$, onde:

- $N = N_s \cup N_i \cup N_t$ representam o conjunto de nós (por exemplo: componentes), em que N_s é o conjunto de nós iniciais (*source*), N_t é o conjunto de nós destino (*target*) e N_i denota o conjunto de nós intermediários, $N_s \cap N_i = N_s \cap N_t = N_i \cap N_t = \emptyset$;
- $A \subseteq (N_s \times N_i) \cup (N_i \times N_t) \cup (N_i \times N_i) = \{(a,b) \mid a \neq b\}$ denota o conjunto de arestas (por exemplo: as conexões dos componentes).
- $w : A \rightarrow \mathbf{R}^+$ é uma função que associa pesos as arestas (o valor associado a aresta (j, k) é adotado para distribuição da energia associada ao nó j para o nó k de acordo com a relação $w(j,k)/\sum_{i \in j^\bullet} w(j, i)$, onde j^\bullet é o conjunto de nós que partem do nó j);
- $f_d : N \rightarrow \begin{cases} \mathbf{R}^+ & \text{se } n \in N_s \cup N_t, \\ 0 & \text{caso contrário;} \end{cases}$

é uma função que associa a cada nó o calor a ser extraído (considerando modelos de resfriamento) ou a energia a ser suportada (considerando modelos elétricos);

- $f_c : N \rightarrow \begin{cases} 0 & \text{se } n \in N_s \cup N_t, \\ \mathbf{R}^+ & \text{caso contrário;} \end{cases}$

é a função que associa cada nó com sua respectiva capacidade máxima de energia;

- $f_p : N \rightarrow \begin{cases} 0 & \text{se } n \in N_s \cup N_t, \\ \mathbf{R}^+ & \text{caso contrário;} \end{cases}$

é uma função que associa cada nó (um nó representa um componente) com seu preço de venda;

- $f_\eta : N \rightarrow \begin{cases} 1 & \text{se } n \in N_s \cup N_t, \\ 0 \leq k \leq 1, k \in \mathbf{R} & \text{caso contrário;} \end{cases}$

é uma função que associa cada nó com sua eficiência energética.

O modelo EFM adota uma abordagem de primeira profundidade para percorrer o modelo (grafo) quando computa suas métricas. Além disso, um vetor que contém todos os nós (componentes) do modelo é adotado, para associar a cada nó $i \in N_i$ a energia de corrente elétrica que flui através de cada componente.

2.6.1 Grafo Acíclico Dirigido

Um grafo pode ser descrito como uma estrutura constituída por dois elementos: arco e vértice [20]. Nas últimas décadas, a relação entre algoritmos e grafos foi estudada com afinco. Em geral, a área de algoritmos e grafos pode ser caracterizada como uma área cujo interesse primário é o de resolver problemas de grafos utilizando-se de algoritmos, tendo-se em mente uma preocupação com o desempenho dessa solução. Nesse estudo, o processo se repete, uma vez que temos um problema em um grafo e propomos um algoritmo para solucioná-lo.

Grafos são classificados como: completo, simétrico, cíclico, dirigido, etc. [20]. Aqui são adotados os conceitos dos grafos acíclicos dirigidos (GAD), uma vez que se encaixam com a modelagem proposta para avaliação do fluxo elétrico EFM.

Um grafo acíclico dirigido $D(V,E)$ é composto por um conjunto finito e não vazio de vértices (V) e um conjunto de arestas (E - "edges") [11]. Assim, em um grafo dirigido,

cada arco (v, w) possui apenas uma direção, neste caso de v para w . Em um GAD, os vértices inicial e final não podem ser o mesmo para nenhum subcaminho no grafo.

Para mais detalhes sobre os grafos, o leitor é redirecionado a [20] [11].

2.6.2 Custo

A capacidade de avaliação do custo em um centro de dados é essencial para uma análise do retorno do investimento, além de outras decisões de negócios. No ambiente Mercury o custo é dividido entre o de aquisição e o operacional. O custo de aquisição é o recurso financeiro necessário para comprar a infraestrutura. O custo operacional corresponde ao custo da operação do centro de dados por um período de tempo, considerando-se o consumo de energia, custo, período de avaliação, e a disponibilidade do centro de dados ou da nuvem.

O custo operacional é utilizado como métrica para avaliar financeiramente o ambiente em questão, uma vez que se relaciona com disponibilidade e consumo de energia. É computada pela Equação 2.19. Adicionalmente, é importante frisar que outros custos podem ser incrementados a esta equação, a exemplo do custo de manutenção.

$$OpCusto = P_{Entrada} \times C_{Energia} \times T \times (A + \alpha(1 - A)) \quad (2.19)$$

onde $OpCusto$ é o custo operacional; $P_{Entrada}$ é a eletricidade consumida; $C_{energia}$ é o custo da energia; T é o período considerado; A é a disponibilidade e α é o fator adotado para representar a quantidade de energia que continua a ser consumida após a falha de um componente.

3

Algoritmo de Distribuição de Carga Elétrica

Esse capítulo apresenta o Algoritmo de Distribuição de Carga Elétrica (ADCE) proposto, bem como suas principais funcionalidades e aplicabilidade.

3.1 Contextualização

Não existe, atualmente, um algoritmo que distribua o fluxo elétrico em uma rede, baseada nas eficiências de cada componente da rede. A distribuição deste fluxo fica a cargo do projetista do ambiente ou da configuração padrão da ferramenta utilizada. Portanto, havendo interação humana na escolha da distribuição desse fluxo, torna-se possível a existência de casos de falhas, ou se a configuração ficar sob responsabilidade da ferramenta seguirá sua própria metodologia para distribuição. Tal imperfeição (humana ou via configuração padrão da ferramenta) pode ensejar um maior consumo elétrico, acarretando maiores custos às empresas.

A ferramenta utilizada nesta dissertação foi o Mercury, que faz a distribuição do fluxo elétrico na rede de forma proporcional a quantidade de nós do modelo EFM. Visando solucionar o problema, foi concebido um algoritmo para distribuir o fluxo elétrico em uma rede de forma heurística (ADCE), onde as eficiências dos equipamentos definem o melhor percurso do fluxo na rede.

A Figura 3.1, apresenta um modelo EFM simples. Este modelo considera que o fluxo será escolhido pela configuração padrão da ferramenta Mercury, informado através dos pesos das arestas. Os pesos de 0.5 representam metade do fluxo do componente SDT1 para cada UPS, logo, a distribuição do fluxo é feita de forma proporcional a quantidade de nós pais. Caso a Figura 3.1 apresentasse três componentes do tipo UPS (UPS1, UPS2 e UPS3) conectados ao componente SDT1, o fluxo seria dividido em três partes iguais,

sendo os pesos de 0.33 para cada aresta.

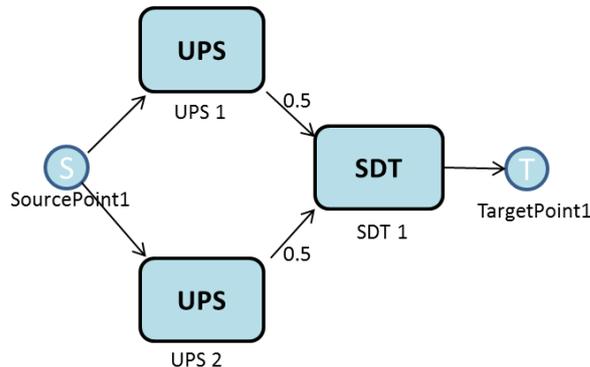


Figura 3.1: Parâmetros do modelo EFM

Em muitas situações, não se considera a eficiência energética dos equipamentos, como é o caso do exemplo da figura, no qual o fluxo é dividido igualmente, sem se considerar as eficiências dos UPSs, incorrendo, possivelmente, em uma distribuição inadequada da energia.

3.2 O Algoritmo

O Algoritmo de Distribuição de Carga Elétrica (ADCE) é proposto para minimizar o consumo elétrico do modelo EFM [16]. O (ADCE) foi inspirado no algoritmo de marcação de fluxo máximo de Ford-Fulkerson, que computa o fluxo máximo possível em uma rede de fluxos [31]. A modelagem de fluxo elétrico (*Energy Flow Model -EFM*) é representada por um grafo, acíclico e dirigido, onde os componentes são modelados como vértices e as respectivas ligações correspondem às arestas. O fluxo a ser transportado por cada componente é definido pelas arestas e limitado pela capacidade dos vértices.

O algoritmo inicia transcorrendo o grafo à busca do melhor caminho entre dois pontos específicos. Se um determinado caminho não possuir capacidade para suportar todo o fluxo exigido, então será transmitida a capacidade máxima possível deste fluxo por este caminho e o residual será redirecionado para ser transferido em uma nova iteração, à procura de outro caminho. Caso não exista outro caminho capaz de transportar o fluxo residual, mediante as restrições de capacidade dos componentes, não será possível a transmissão desse fluxo através do grafo.

A busca de primeira prioridade (*Priority First Search - PFS*) é adotada como método para seleção do caminho entre os nós. A PFS tradicional escolhe o melhor caminho

através das maiores capacidades entre os nós do grafo [22]. Porém, essa procura foi adaptada no ADCE para escolher o caminho através das maiores eficiências entre os nós do grafo, respeitando os limites de capacidade. Para maiores informações sobre o algoritmo de Ford-Fulkerson, o leitor é encaminhado a [31].

A Figura 3.2 demonstra a configuração dos parâmetros do modelo EFM. O fluxo a ser distribuído (chamado energia elétrica demandada) é um elemento do modelo EFM. A demanda é inserida no vértice destino (TargetPoint1 da Figura 3.2) e será um critério de parada do algoritmo. Enquanto houver uma demanda energética não alocada, um caminho será avaliado para transmiti-la. A escolha é feita entre os vértices pertencentes ao caminho mais eficiente, sendo o vértice com a menor capacidade, presente nesse caminho, o que limita o fluxo a ser transmitido.

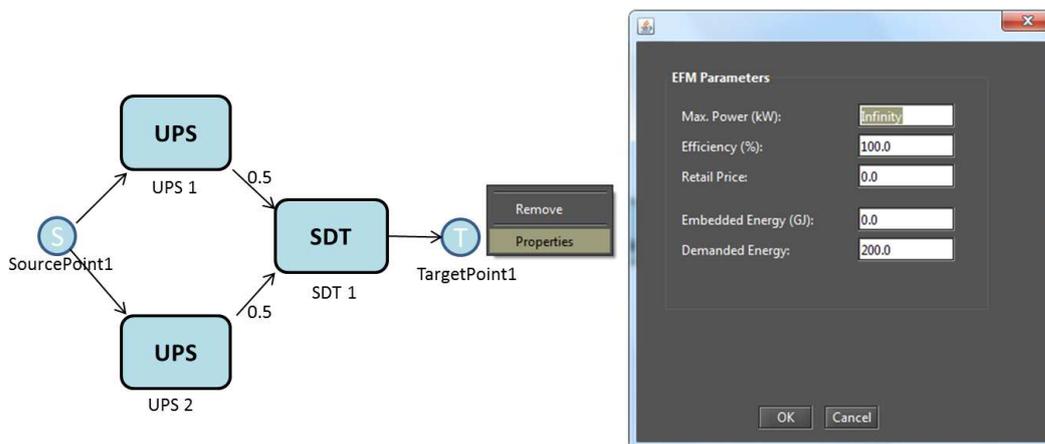


Figura 3.2: Parâmetros do modelo EFM

Ao final de cada iteração, o valor a ser transmitido pelo caminho escolhido é subtraído da demanda. Cada vértice pertencente ao caminho terá o valor da variável responsável por armazenar o seu fluxo aumentado e sua capacidade de armazenar potência decrementada pelo valor transmitido. Todas as alterações são aplicadas a uma cópia do modelo EFM, não sendo realizada qualquer alteração no modelo original.

Foram adotados os seguintes valores para os vértices de origem e destino do grafo, representados na Figura 3.2 por *SourcePoint1* e *TargetPoint1*: para a potência máxima suportada, o valor de infinito; para a eficiência, o valor de 100% e para o preço e a exergia, os valores de 0.0. O campo de demanda é utilizado apenas nos vértices *TargetPoint*.

Para uma melhor compreensão da distribuição, é necessário que se entenda como o melhor caminho é escolhido dentro do modelo, ou seja, baseado nas eficiências de cada

componente.

3.2.1 Caminho Válido

Nessa subseção será ilustrada a função que determina a validade ou não de um caminho. Na Figura 3.3, foram destacadas duas informações com relação a alguns vértices fundamentais para a compreensão da validade do caminho: capacidade e eficiência dos componentes (ACSource), representadas pelas letras C e E, respectivamente. Estas letras representam as variáveis manipuladas pelo algoritmo para determinação e distribuição do fluxo. O valor atual da variável que armazena a demanda do modelo também é ressaltado no vértice de destino (TargetPoint1). Nesse ponto é identificada a demanda a ser distribuída pelo modelo (TargetPoint1 de 1200kW), a capacidade e a eficiência dos vértices ACSource 1 (capacidade de 700kW e eficiência de 95%) e ACSource 2 (capacidade de 700kW e eficiência de 90%).

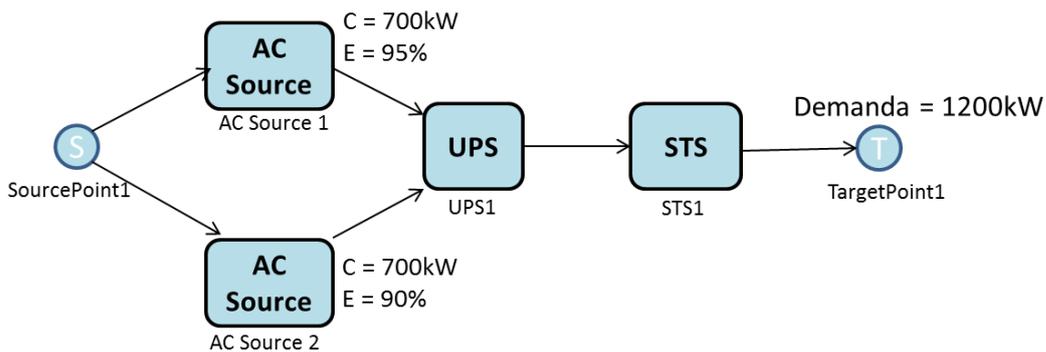


Figura 3.3: Exemplo de modelo EFM

Na Figura 3.4 resalta-se o melhor caminho válido, de acordo com as eficiências dos componentes do modelo. Para fins didáticos e, considerando a primeira iteração do algoritmo, a cor verde foi atribuída às arestas que identificam o caminho. O maior valor possível é distribuído (700kW, que é a menor capacidade dos componentes do caminho), acrescido esse valor às arestas (como fluxo corrente na aresta) e subtraindo-o das capacidades dos vértices. O valor distribuído é também subtraído da demanda. Como resultado desses eventos, as variáveis do modelo são alteradas, veja a Figura 3.4.

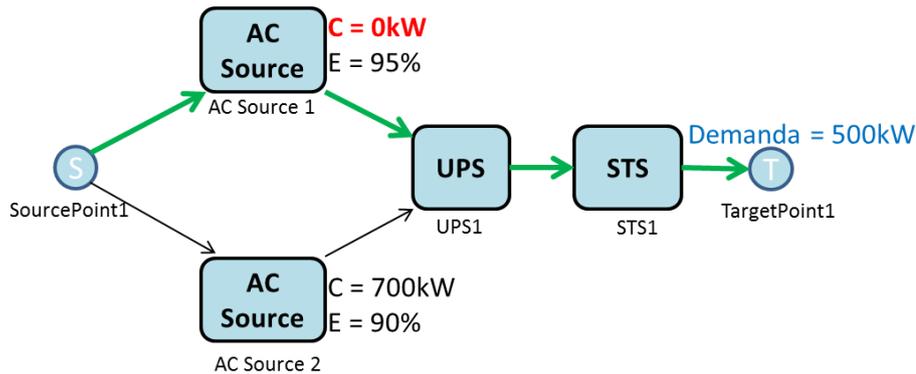


Figura 3.4: Exemplo de modelo EFM - 1ª Iteração

Na segunda iteração do algoritmo, o caminho encontrado na primeira não é mais considerado válido, devido à restrição da capacidade de fluxo suportada pelo componente ACSource1, ou seja, o componente ACSource 1 suporta uma capacidade de 0kW. Esse caminho encontra-se, destacado em vermelho, na Figura 3.5. Assim, o melhor caminho válido, de acordo com as eficiências dos componentes, está destacado na cor verde.

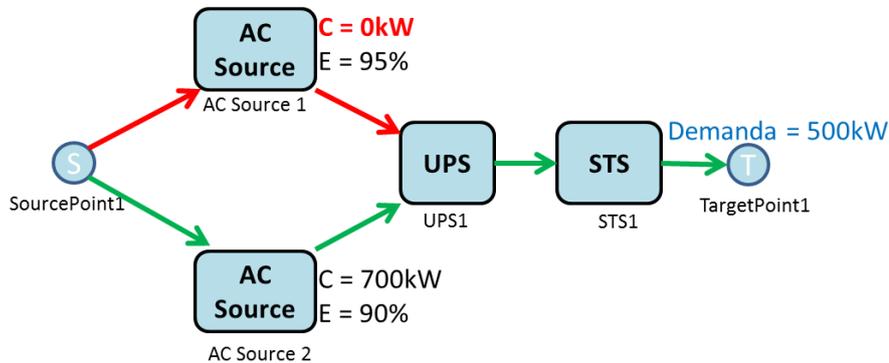


Figura 3.5: Exemplo de modelo EFM - 2ª Iteração

As operações de incremento e decremento das variáveis do modelo que lidam com o fluxo transcorrido ocorrem de forma semelhante à descrição da primeira iteração. Após a segunda iteração, a variável que armazena o valor da demanda do modelo atinge o valor zero (critério de parada da busca por novos caminhos), não havendo, dessa forma, mais fluxo a ser distribuído, embora o modelo ainda seja capaz de suportar a distribuição de 200kW pelo componente ACSource 2.

A Figura 3.6 apresenta o resultado das variáveis destacadas no modelo após a execução da parte do algoritmo que escolhe um caminho válido. Neste ponto, a distribuição da

demanda pelos melhores caminhos, de acordo com as eficiências dos componentes, chega ao final.

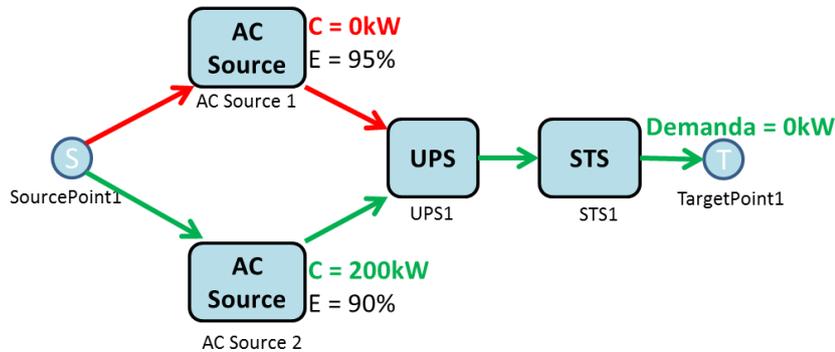


Figura 3.6: Exemplo de modelo EFM - Final

De posse do conhecimento acerca dos procedimentos adotados para escolha do melhor caminho entre os componentes do modelo, será apresentado o algoritmo ADCE, composto por três fases:

- Fase I - Inicialização
- Fase II - Kernel
- Fase III - Distribuição dos Pesos

3.2.2 FASE I - Inicialização do ADCE

O Algoritmo 1 (método de inicialização) é um pseudo-código que representa a leitura do modelo EFM, bem como a inicialização das variáveis, as chamadas ao *kernel* do ADCE e a atualização dos pesos das arestas do modelo. O número de chamadas ao *kernel* corresponde ao número de nós destinos (*target nodes*), presentes no modelo EFM G , configurado como parâmetro do algoritmo 1.

O primeiro passo do algoritmo é fazer uma cópia do modelo EFM original G (linha 1). As futuras operações serão realizadas nessa cópia, que é representada pela variável R . Posteriormente, um vetor chamado cca (capacidade de potência acumulada), é empregado para associar a cada nó um valor que expressa a capacidade de potência acumulada no vetor.

Algorithm 1 *Inicializar ADCE (G)*

```

1:  $R = G$ 
2: for  $i \in N$  do
3:    $cca_i = 0$ 
4: end for
5: for  $a \in A$  do
6:    $weight_a = 0$ 
7: end for
8: for  $n \in N_t$  do
9:    $R = ADCE(R, demanded, n)$ 
10: end for
11:  $setUpdteWeight(R)$ ;
12: return  $R$ 

```

Nas linhas 2 a 4 há uma inicialização da capacidade de potência acumulada de cada nó do modelo, atribuindo o valor zero a cada uma delas. Da mesma forma, existe outro vetor, chamado *weight* que associa a cada aresta um valor que expressa o fluxo nestas arestas, como demonstrado nas linhas 5 a 7, onde tais variáveis também recebem o valor zero.

Uma estrutura de repetição é inicializada na linha 8, na qual cada nó de destino será um índice dessa iteração ($n \in N_t$). A repetição se dá para cada nó (n) pertencente ao conjunto dos nós *Target* (N_t), ou seja, dentro dos nós do modelo há um conjunto que representa especificamente aqueles de destino (*Target*). Em seguida, o modelo R é alterado pela chamada do método ADCE, que será explicado em detalhes na fase seguinte. Nessa chamada, é passado como parâmetro o modelo R , o fluxo demandado pelo nó *target* em questão e o próprio nó *target*.

O ADCE foi desenvolvido com a capacidade de avaliar modelos compostos por mais de um nó destino (*target node*), ou seja, no modelo EFM é possível adicionar mais de um destino no mesmo modelo. Por exemplo, imagine o caso onde há duas nuvens (*clouds*) associadas a uma mesma rede elétrica, onde cada uma das nuvens consuma valores diferentes de energia. Logo, seria possível a existência de dois nós de destino em um mesmo modelo. A linha 11 é responsável pela realização da atualização dos pesos das arestas do modelo R . Este método será detalhado na Fase III.

3.2.3 FASE II - Kernel do ADCE

O método ADCE (Algoritmo 2) é um pseudo-código que simboliza o núcleo do algoritmo de distribuição de carga elétrica. Recebe como parâmetros o modelo EFM, representado pela letra R , o fluxo demandado pelo nó destino (*demanded*) e o próprio nó destino (n).

O ACDE inicia associando a uma variável local *dem* a energia demandada (*demanded*) (linha 1). Em seguida, é chamada uma função que verifica se existe um caminho válido do nó destino (*n*) ao nó origem (*s*), ambos fornecidos pela variável *R*.

Um caminho válido é um caminho de um nó a outro, no qual as capacidades elétricas de todos os componentes desse caminho são respeitadas. A variável *P* é um vetor que armazena os elementos de um caminho válido (linha 3). Uma vez verificada a existência de um caminho válido, a variável *fp* (fluxo possível) é associada ao valor da demanda, que no pseudo-algoritmo é representado pela variável *dem* na linha 4.

Em seguida, entre as linhas 5 a 7, uma estrutura de repetição é utilizada para se obter o menor fluxo possível no vetor *P*. Uma função chamada *getMinimumCapacity* é responsável por esta resposta e possui como parâmetros o fluxo possível (*fp*) e o resultado da subtração da capacidade do fluxo e sua potência acumulada ($capacity_{(i)} - cca_{(i)}$), que reflete o fluxo ainda suportado pelo nó *i*.

Algorithm 2 ACDE (*R, demanded, n*)

```

1: dem = demanded;
2: while (isPathValid(R, n, s)) & (dem > 0) do
3:   P = getElementsFromValidPath(R, n, s);
4:   fp = dem;
5:   for i ∈ P do
6:     fp = getMinimumCapacity(fp, capacity(i) - cca(i));
7:   end for
8:   for i ∈ P do
9:     ccai = ccai + fp;
10:    addFlowArrow(fp);
11:  end for
12:  dem = dem - fp;
13: end while
14: verifyExceptions(R);
15: return R

```

O próximo passo do algoritmo será uma nova iteração utilizando o vetor *P* (linhas 8 a 10), que representa um caminho válido. Neste ponto é realizada uma atualização da potência acumulada em cada nó, somando-se o valor do fluxo possível à potência já acumulada pelo nó. Em seguida, o valor da aresta onde o fluxo está transitando é atualizado, incrementando-se o valor do fluxo possível à aresta, ou seja, o valor que se encontra transitando, no momento, por este nó, é adicionado à aresta correspondente. A variável *dem*, utilizada para identificar a demanda elétrica da arquitetura avaliada, é decrementada pela variável fluxo possível (linha 12), isto é, subtrai-se da demanda o fluxo transmitido por certo caminho, uma vez que esta variável é critério de parada do

algoritmo. Caso esse valor atinja zero (linha 2), significa que não há mais fluxo a se transmitir. Todos esses passos, desde a linha 2, são repetidos enquanto houver fluxo a ser distribuído ($dem > 0$) e um caminho válido. A linha 13 encerra essa iteração. Na linha 14 uma função chamada *verifyExceptions(R)* é utilizada para verificar as exceções possíveis a esta lógica no modelo R, como por exemplo valor negativo na variável *dem*, arquitetura não suporta o fluxo demandado, etc. Na linha 15, é retornada, na variável R, uma cópia do modelo EFM com o fluxo distribuído de forma otimizada.

O próximo passo é fazer a operação de transformação do fluxo corrente nos vértices, em pesos para as arestas do modelo, consoante demonstrado na Fase III do algoritmo.

3.2.4 FASE III - Distribuição de Pesos do ADCE

O Algoritmo 3 é um pseudo-código que simboliza a distribuição dos pesos das arestas. Para os usuários da ferramenta que utilizam o ADCE, tal distribuição é fundamental, pois informa, de maneira clara o peso em cada aresta do modelo, visando a melhoria dos valores das métricas computadas pela ferramenta.

Na Fase III, o algoritmo inicia sua execução recebendo como entrada um modelo EFM, já com a distribuição do fluxo feita pela fase anterior, de acordo com as eficiências dos dispositivos. Conforme descrito na Fase I, todas as arestas do modelo são inicializadas com o valor 0, logo, nessa fase, assume-se que havendo algum valor na aresta, haverá fluxo percorrendo-a. O referido fluxo foi inserido na Fase II (linha 10 do Algoritmo 2).

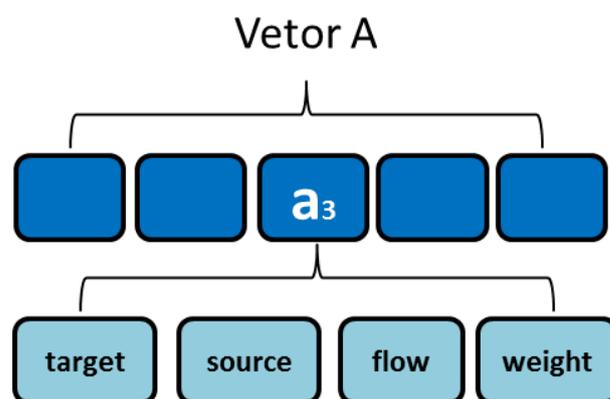


Figura 3.7: Descrição da composição do Vetor A

Ao iniciar, o algoritmo captura todas as arestas do modelo R através do método *getArrows(R)* e as associa a um vetor chamado de A (linha 1). A Figura 3.7 representa

a organização do vetor A , onde são armazenados todos os vértices do modelo R . Cada aresta dentro desse vetor A é chamada de a_i , onde o i representa a quantidade de arestas do modelo. No exemplo é demonstrada a composição de cada aresta (semelhante ao vértice a_3), onde as mesmas possuem um nó *target* (destino), um nó *source*(origem), uma variável para armazenar o fluxo (*flow*) e outra variável para armazenar o peso (*weight*).

Na linha 2 do Algoritmo 3 é iniciada uma iteração, onde serão percorridos todos os valores desse vetor A , ou seja, todas as arestas do modelo R . Uma estrutura condicional é utilizada para verificar se há fluxo na aresta corrente a (linha 3). Caso a resposta seja negativa, o algoritmo passa para a próxima aresta a , dentro do vetor A . Caso a resposta seja positiva, a execução das operações presentes dentro da estrutura condicional deverá ter continuidade (linhas 4 e 5).

O próximo passo é calcular o valor do peso a ser atribuído a cada aresta, dividindo-se o valor do fluxo do nó a ($a.getFlow$) pelo fluxo contido na variável *target* de a ($a.getTarget.cca$) (linha 4), que armazena o nó dentro do modelo R correspondente ao *target* de A . Vale ressaltar que, toda aresta possui um nó *target* (destino) e um nó *source* (origem).

Finalmente, o valor do peso é inserido na variável *weight* (linha 5), exibida no modelo EFM, visível ao usuário da ferramenta Mercury.

Algorithm 3 Distribuir Pesos (R)

```

1:  $A = getArrows(R)$ ;
2: for  $a \in A$  do
3:   if  $a.flow > 0$  then
4:      $Weight = a.getFlow/a.getTarget.cca$ ;
5:      $a.setWeight(Weight)$ ;
6:   end if
7: end for

```

Com a finalidade de facilitar o entendimento do algoritmo, adiante seguem dois exemplos da execução resumida do ADCE.

3.3 Exemplo 1

Esta seção exemplificará a execução do ADCE em um modelo EFM simples. A Figura 3.8 demonstra o modelo utilizado para este exemplo, onde os valores ressaltados representam a capacidade (C), eficiência (E) e potência acumulada (cca) dos vértices, bem como o fluxo (F) e o peso (W) das arestas. Com exceção do peso da aresta, essas informações

não são visíveis ao usuário no modelo original. Por essa razão estão destacadas, visando facilitar a explicação.

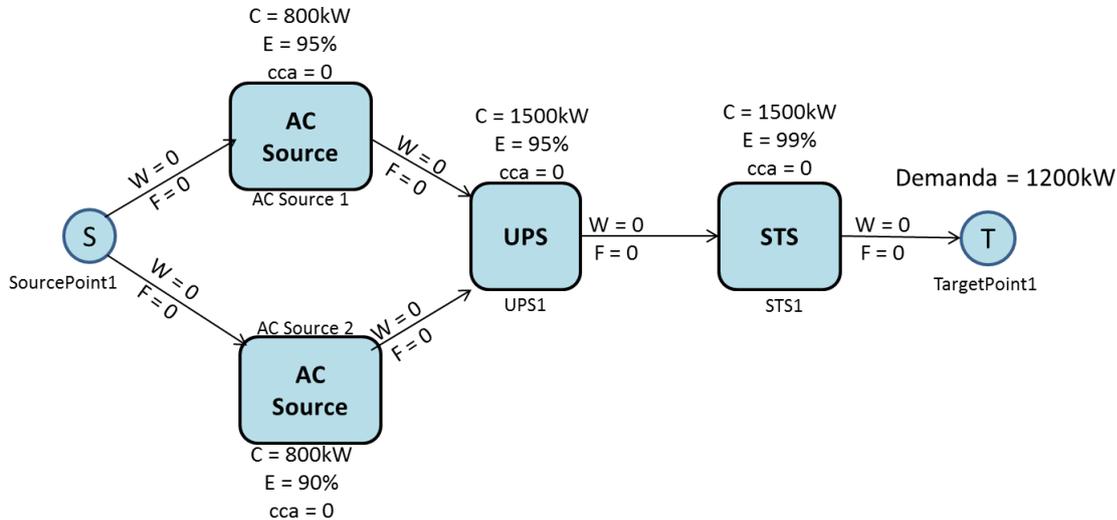


Figura 3.8: Exemplo EFM

Na primeira fase do algoritmo as variáveis são inicializadas, como representado na figura 3.8. O *kernel* é chamado para efetuar a distribuição do fluxo, respeitando os limites de capacidade e a escolha do melhor caminho válido, como descrito na Seção 3.2.1. Após o encerramento da segunda fase, o modelo se encontrará como descrito na Figura 3.9, com os valores de fluxo preenchidos em cada aresta.

Neste ponto tem início a terceira fase do algoritmo, onde é realizada a conversão do fluxo em pesos para as arestas, facilitando o trabalho do projetista do ambiente. Como descrito na fase 3, o peso é calculado dividindo-se o valor do fluxo da aresta a ($a.flow$), pela potência acumulada no vértice considerado como destino desta aresta a ($a.getTarget.cca$).

É importante ressaltar que na varredura do grafo, o vértice considerado como destino é aquele apontado pela aresta. Ademais, a varredura do modelo se inicia pelo nó *Target* e termina no nó *Source*.

Por exemplo, na aresta localizada entre os componentes que representam o UPS e STS (ressaltada em verde na Figura 3.9), o fluxo da aresta a é $1212,12\text{kW}$ e a potência acumulada (cca) do *target* desta aresta (STS) é $1212,12\text{kW}$. Logo, o valor resultante da operação ($Weight = a.getFlow/a.getTarget.cca$) é 1, que equivale ao peso daquela aresta. Sempre que um vértice possuir apenas um vértice apontando para ele (ou seja,

apenas um nó pai), o valor do peso será 1.

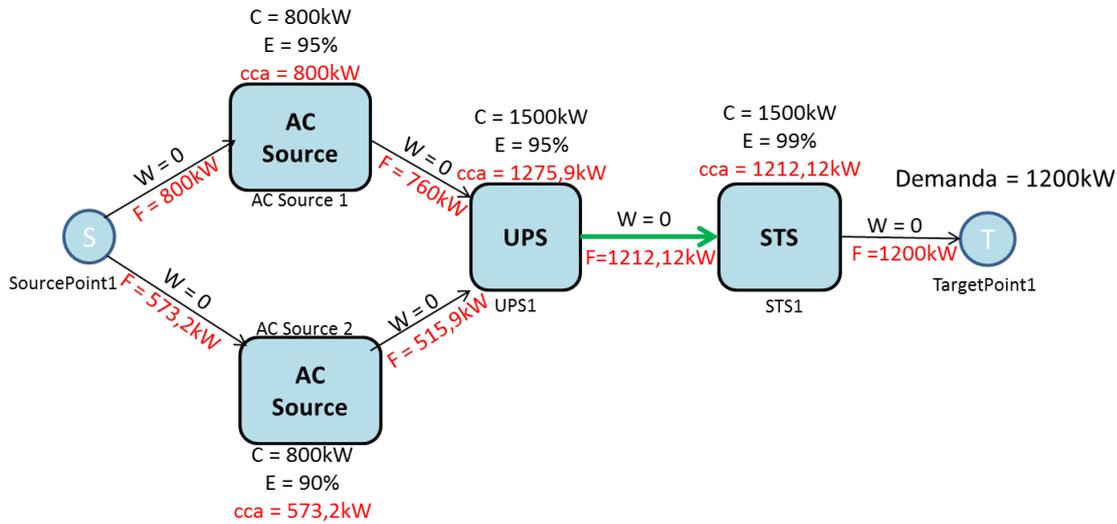


Figura 3.9: Exemplo EFM - Fase 2 concluída

No caso dos componentes *AC Source 1* e *AC Source 2*, em relação ao UPS, o valor do peso não será 1, embora o procedimento seja o mesmo. Como o fluxo do UPS está dividido entre os dois *AC Sources* de forma desigual, o algoritmo informa uma distribuição heurística do fluxo através dos pesos. Considerando-se a aresta que conecta os componentes *AC Source 1* ao UPS, o fluxo armazenado é de 760kW e o valor da potência acumulada no vértice *target* dessa aresta é $1275,9\text{kW}$. Portanto, o peso da aresta é $0,59$. Idêntico procedimento é adotado para o cálculo do peso da aresta que liga o *AC Source 2* ao UPS, resultando em um valor de $0,41$.

A Figura 3.10 apresenta a configuração final do modelo, com todos os pesos atribuídos às arestas, ressaltados na cor azul. Destaca-se que todas as variáveis, como fluxo, capacidade, potência acumulada e demanda, estão embutidas na codificação do algoritmo e não são exibidas ao usuário do aplicativo. Com relação ao peso, houve opção por não exibi-lo quando igual a 1. Consequentemente, será exibido se o valor for diferente de 1.

A seguir será apresentado um segundo modelo, mais complexo, por intermédio do Exemplo 3.4, onde fica mais evidente a necessidade de um algoritmo para distribuir o fluxo baseado nas eficiências dos componentes do modelo.

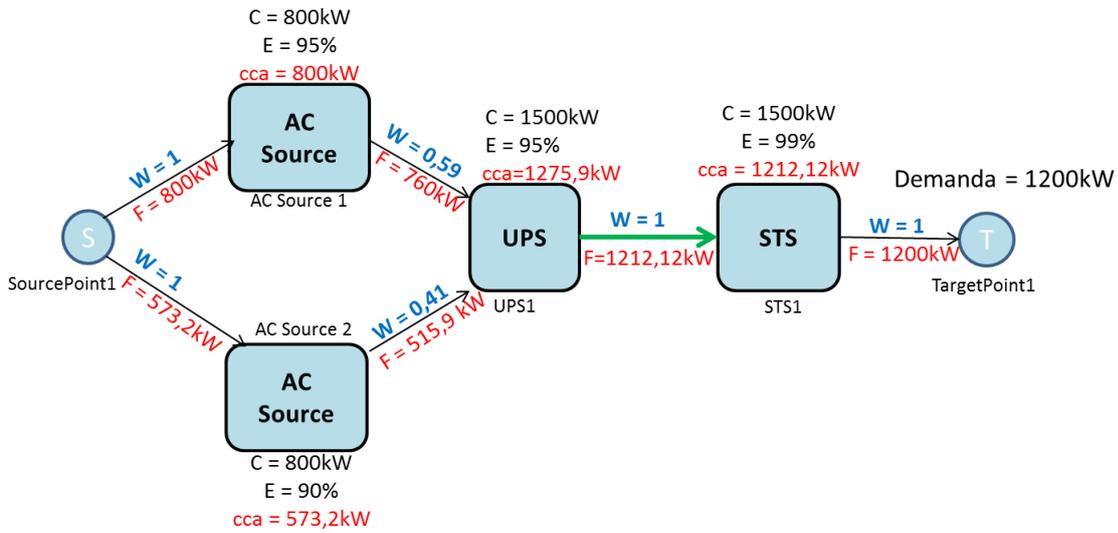


Figura 3.10: Exemplo EFM - Concluído

3.4 Exemplo 2

No segundo exemplo da execução do ACDE, adota-se um modelo um pouco maior. A Figura 3.11 representa um modelo EFM de uma arquitetura elétrica de um centro de dados. Este modelo representa uma variação de uma arquitetura elétrica real de um centro de dados presente nos laboratórios da HP, em Palo Alto, Estados Unidos [48]. Os retângulos simbolizam equipamentos elétricos e foram utilizados para simplificar o exemplo. Todos os pesos das arestas estão, de modo padrão, configurados para 1 (ocultos nas arestas da Figura 3.11).

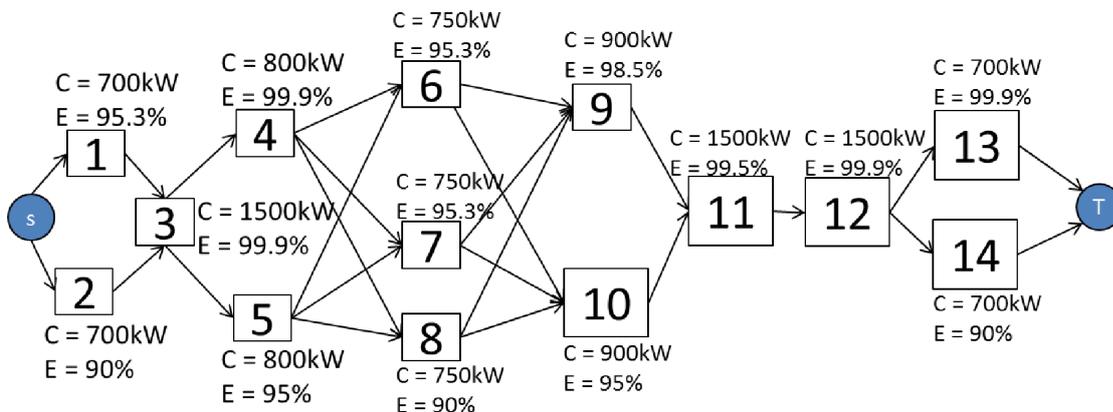


Figura 3.11: Modelo EFM.

Da mesma forma que o exemplo anterior, o fluxo é distribuído e os pesos atribuídos às arestas. A Figura 3.12 exibe o modelo EFM após a execução do ADCE. É importante notar que os pesos nas arestas mudaram significativamente, melhorando o fluxo elétrico, em decorrência da distribuição de pesos mais adequados a redução do consumo elétrico do modelo.

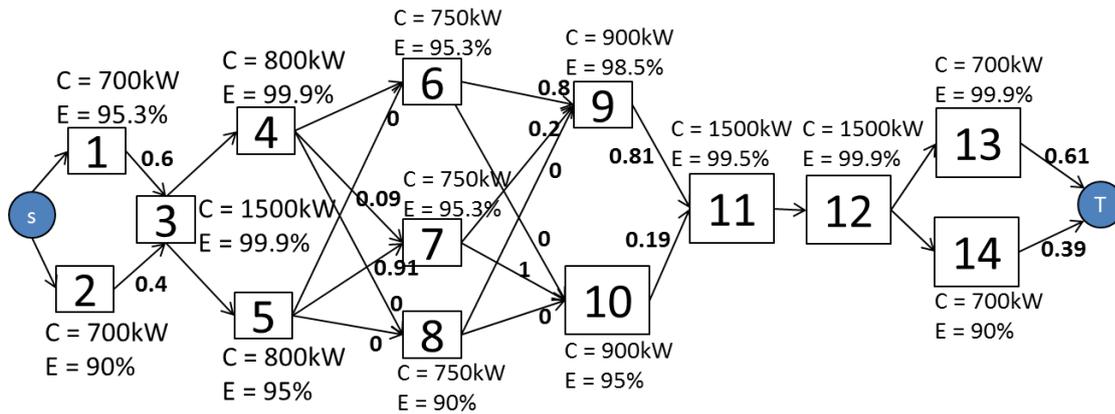


Figura 3.12: Modelo EFM com a execução do ADCE.

Diferente do exemplo anterior, é perceptível a maior quantidade de arestas a terem seus pesos alterados, visando alcançar uma melhor distribuição do fluxo. Uma comparação com algumas métricas calculadas através do Mercury, usando a configuração padrão e após a aplicação do ADCE, é demonstrada a seguir, para comprovar a aplicabilidade do algoritmo.

A Tabela 3.1 apresenta um resumo dos resultados obtidos após aplicação do ADCE no modelo. A coluna “*Conf. Padrão*”exibe os resultados obtidos utilizando a configuração padrão da ferramenta Mercury, onde o fluxo é dividido de forma proporcional entre as arestas que possuem mais de um nó pai, enquanto a coluna “*Com ADCE*”exibe os resultados das métricas com a aplicação do ADCE. A coluna “*Melhora*”demonstra a melhoria, em porcentagem, das métricas em relação aos resultados da configuração padrão e com a aplicação do algoritmo.

Tabela 3.1: Resumo dos resultados sem e com a execução ADCE.

Métrica	Conf. Padrão	Com ADCE	Melhora (%)
Energia de Entrada (kW)	1312.63	1259.64	4.2
Eficiência do Sistema (%)	76.18	79.38	4.2
Custo Operacional (USD)	1264849	1213784	4.2
Exergia Operacional (GJ)	9859.32	8188.11	20.4

A energia necessária para suprir a demanda é reduzida em 4,2%. Consequentemente, os custos e a eficiência do sistema também são melhorados em 4,2%. A métrica de impacto ambiental (exergia) é diminuída em 20,4%.

Essas métricas foram obtidas utilizando o mesmo modelo na ferramenta Mercury. Percebe-se que, quanto maior o modelo, maior a complexidade de distribuição do fluxo. Assim, como a distribuição realizada por um ser humano é passível de erros e as configurações de algumas ferramentas de distribuição de fluxo não são adequadas, a utilização do ADCE melhora as métricas avaliadas.

4

Estudos de Caso

Este capítulo apresenta dois estudos de caso a fim de ilustrar a aplicabilidade do algoritmo proposto de distribuição de carga elétrica (ADCE) nas modelagens de fluxo elétrico (EFM). O ambiente utilizado para avaliar o consumo elétrico foi o ASTRO/Mercury. As subseções detalham a arquitetura elétrica de um centro de dados real e de uma nuvem privada, assim como apresenta os modelos correspondentes (RBD e EFM). Além disso, os resultados da avaliação são apresentados.

4.1 Estudo de Caso 1 - Centro de Dados

O objetivo deste estudo de caso é verificar a aplicabilidade do algoritmo proposto (ADCE) em infraestruturas elétricas de centros de dados. Para isto, algumas infraestruturas elétricas de centros de dados foram avaliadas.

Para cada arquitetura foram calculados os seguintes indicadores: (i) exergia operacional, (ii) custos operacionais (iii) disponibilidade, (iv) o número de 9s (disponibilidade em outro ponto de vista), (v) o tempo de inatividade em horas (*downtime*), (vi) potência de entrada, e (vii) a eficiência do sistema.

Estas métricas foram computadas considerando um período de um ano, em modelos EFM, com a configuração padrão da ferramenta Mercury (dividindo o fluxo proporcionalmente a quantidade de arestas ascendentes a cada vértice) e com a aplicação do algoritmo proposto (ADCE). As Figuras 4.1, 4.2 e 4.3 representam as arquiteturas elétricas dos centros de dados avaliadas neste estudo.

4.1. ESTUDO DE CASO 1 - CENTRO DE DADOS

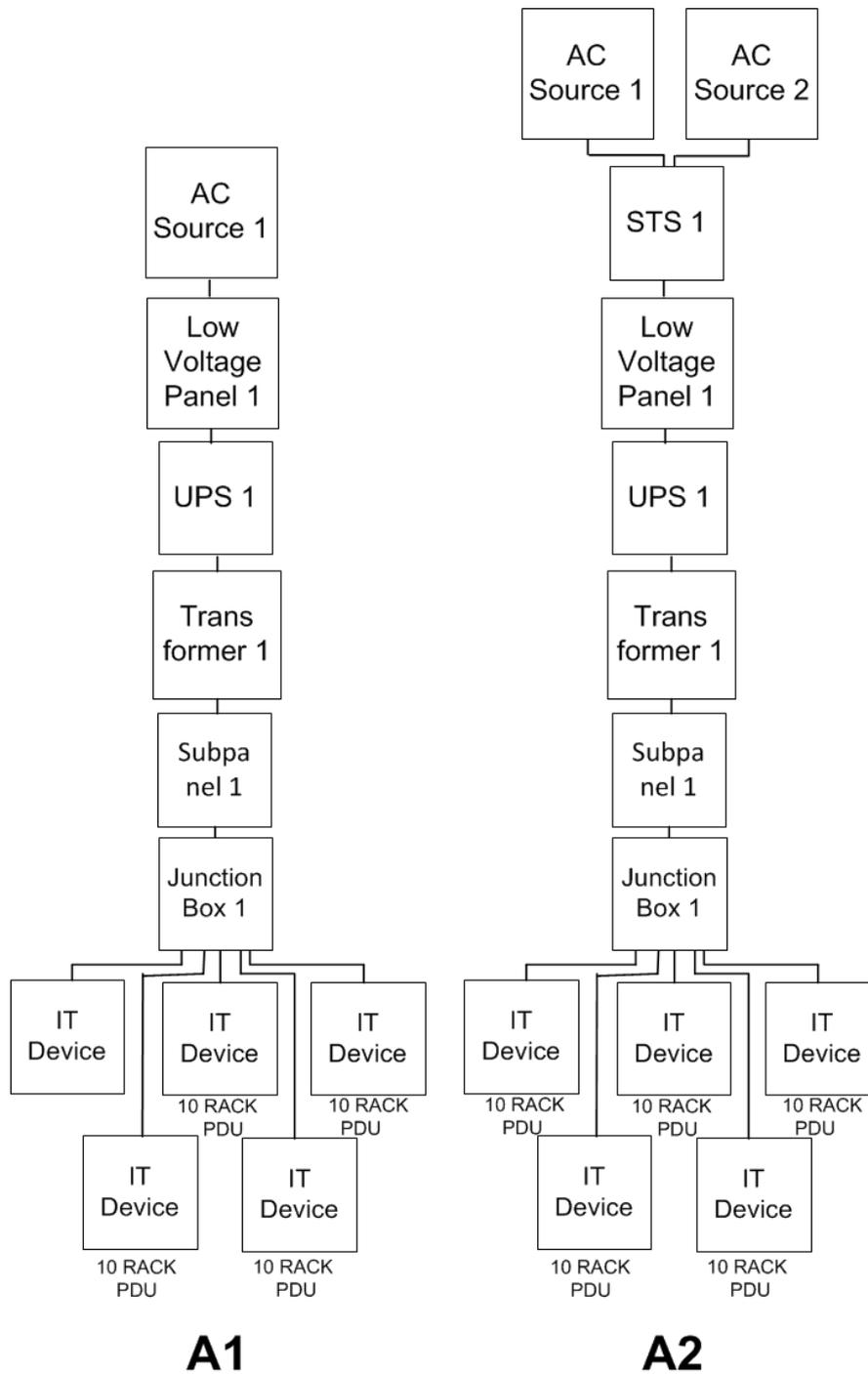


Figura 4.1: Arquiteturas elétricas de centros de dados A1 e A2

4.1. ESTUDO DE CASO 1 - CENTRO DE DADOS

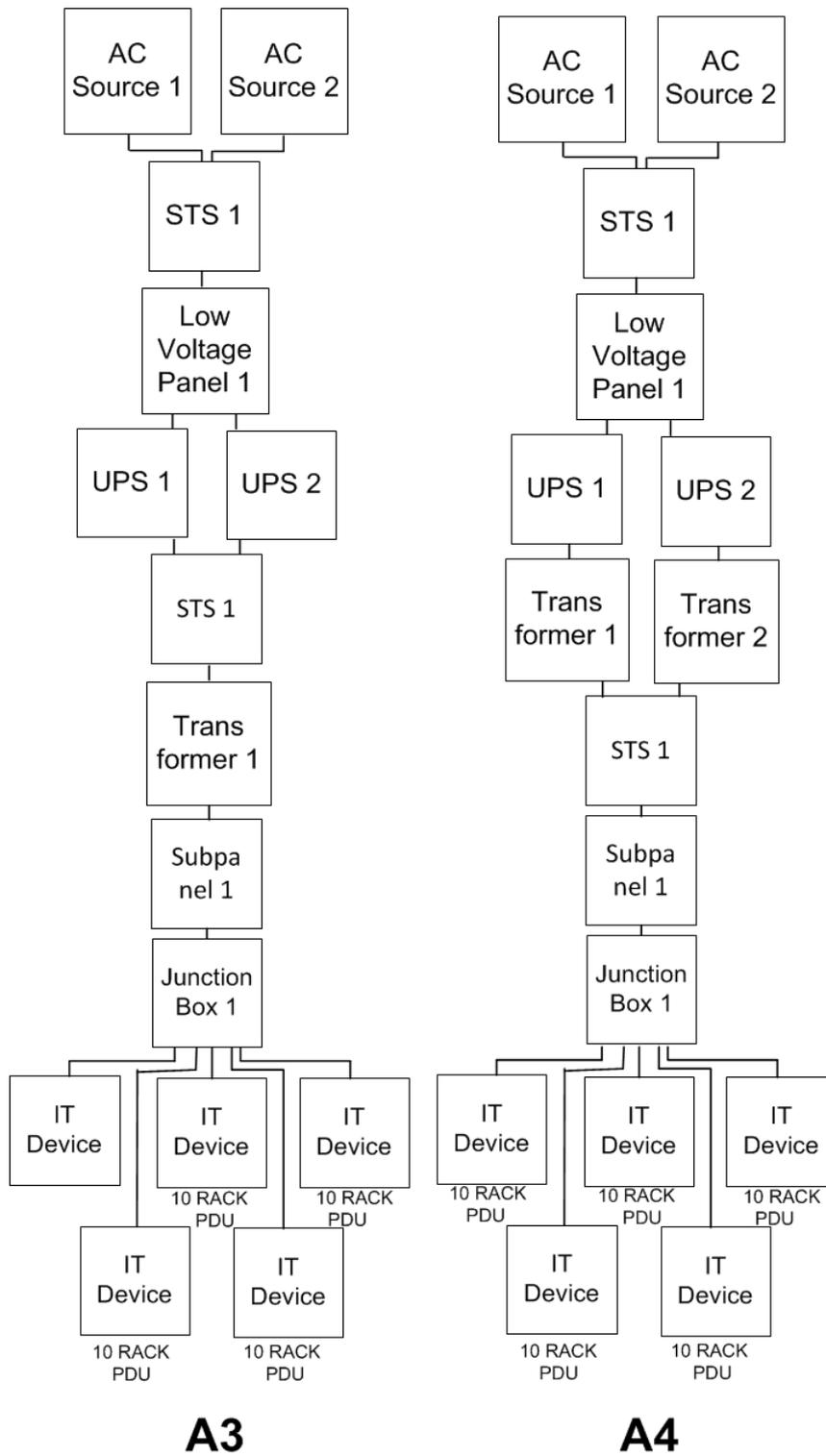


Figura 4.2: Arquiteturas elétricas de centros de dados A3 e A4

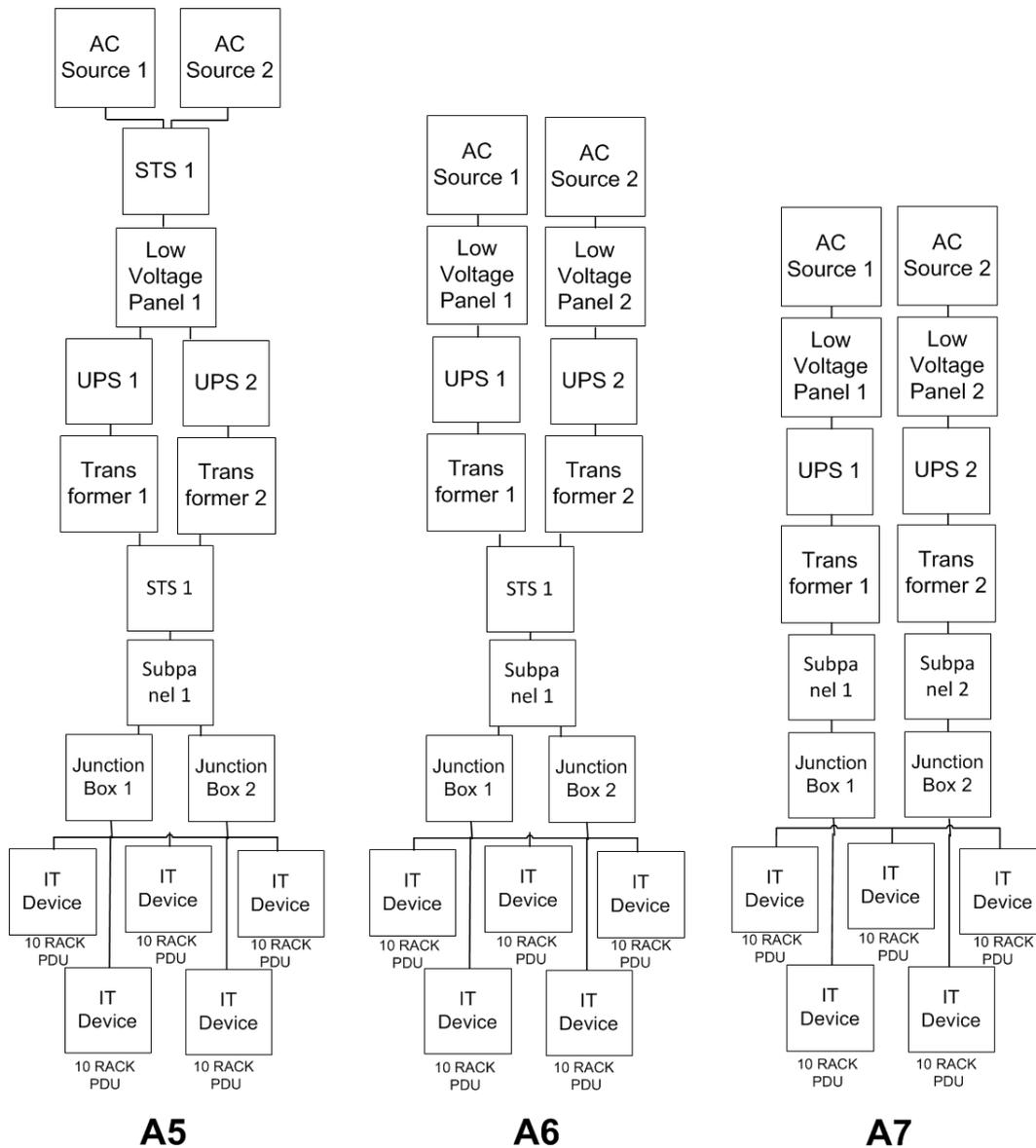


Figura 4.3: Arquiteturas elétricas de centros de dados A5, A6 e A7

4.1.1 Arquiteturas

As arquiteturas analisadas neste estudo de caso são semelhantes àquela presente no laboratório da HP, localizado em Palo Alto nos Estados Unidos e publicada em [48].

A partir da infraestrutura elétrica real representada na Figura 4.1 (Arquitetura A1), outras seis foram criadas, classificadas com a nomenclatura: A2, A3, A4, A5, A6 e A7. A arquitetura A1 apresentada na figura é composta por um *AC Source*, um *Low Voltage Panel*, um *UPS*, um *Transformer*, um *SubPanel*, um *JunctionBox* e cinco *IT Device* (cada

4.1. ESTUDO DE CASO 1 - CENTRO DE DADOS

IT Device é composto por 10 *Racks PDUs*), sendo esta a infraestrutura básica adotada neste estudo.

O índice de importância da confiabilidade e do custo (RCI) foi utilizado para indicar qual a ordem e o componente a se aplicar redundância [14]. Por exemplo, na arquitetura A1, de acordo com o RCI, o componente *ACSource* é o mais indicado a replicação. Portanto, a arquitetura A2 corresponde à A1 considerando o componente *ACSource* replicado. Uma abordagem semelhante foi adotada para propor as demais arquiteturas, até se chegar a A7, onde há uma replicação total da arquitetura A1.

Em geral, na infraestrutura elétrica de um centro de dados o fluxo inicia seu percurso a partir de uma fonte de alimentação (ou seja, *ACSource*), e então flui através de painéis de tensão, fontes de alimentação ininterrupta (UPS), unidades de distribuição de energia (PDUs) (composto por um transformador e um subpainel elétrico), caixas de junção, e, finalmente, para rack de PDUs (unidades de distribuição de energia de rack).

Tabela 4.1: Equipamento, Eficiência e Capacidade

Equipamento	Eficiência(%)	Capacidade(kW)
AC Source 1	95.3	10000
AC Source 2	90	10000
STS 1	99.5	1500
STS 2	98	1500
STS 3	99.5	1500
SDT (or Transformer) 1	98.5	5000
SDT (or Transformer) 2	95	5000
Sub Panel 1	99.9	1500
Sub Panel 2	95	1500
Sub Panel 3	99.9	1500
Low Voltage Panel 1	99.9	1500
Low Voltage Panel 2	95	1500
UPS 1	95.3	5000
UPS 2	90	5000
Junction Box 1	99.9	1500
Junction Box 2	98	1500
Power Strip	95.1	5000

Como detalhado na Tabela 4.1, este estudo de caso emprega dispositivos redundantes com diferentes valores para as eficiências elétricas de cada equipamento. A tabela também apresenta a capacidade máxima de energia que cada dispositivo suporta.

4.1.2 Modelos

Essa seção apresenta a modelagem para arquitetura A5, que corresponde a arquitetura de base A1 com a replicação dos componentes *ACSource*, *UPS*, *Transformer* e *Junction Box*. Os tempos médios de falha (MTTF) e reparo (MTTR) de cada componente necessário ao cálculo de dependabilidade foram obtidos a partir de [5] [55] [43]. A Figura 4.4 ilustra o modelo em RBD correspondente a arquitetura A5.

Os cinco Racks PDUs foram considerados essenciais ao funcionamento do centro de dados e por este motivo modelados em série. Logo, se um deles falha, o centro de dados também falhará.

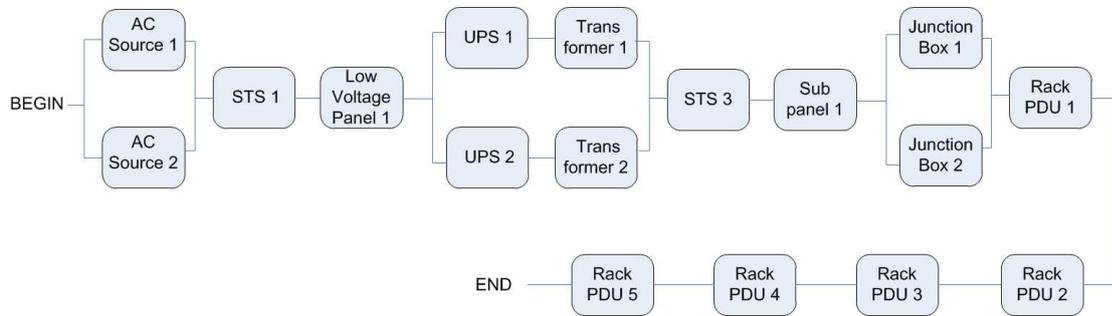


Figura 4.4: Modelo RBD da Arquitetura A5.

A avaliação do modelo RBD permite o cálculo da disponibilidade de acordo com a disposição dos blocos (série-paralelo, pontes, estrelas e disposições em delta). Para a configuração serie-paralelo, derivam-se as funções lógica (Equação 4.1) e estrutural (Equação 4.2), assim como a expressão que permite o cálculo da disponibilidade (Equação 4.3). Essas expressões e funções foram apresentadas no Capítulo 2. A seguir apresentam-se as respectivas do modelo da arquitetura A5, apresentada na Figura 4.4.

Função Lógica:

$$\Psi(S) = ((ACS1) \vee (ACS2)) \wedge (STS1) \wedge (LVP1) \wedge (((UPS1) \wedge (T1)) \vee ((UPS2) \wedge (T2))) \wedge (STS3) \wedge (SP1) \wedge ((JB1) \vee (JB2)) \wedge (RPDU1) \wedge ((RPDU2) \wedge (RPDU3)) \wedge (RPDU4) \vee (RPDU5) \quad (4.1)$$

Função Estrutural:

$$\begin{aligned} \Phi(X) = & (1 - (1 - ACS1) \times (1 - ACS2)) \times (STS1) \times (LVP1) \\ & \times (1 - (1 - (UPS1) \times (T1)) \times (1 - (UPS2) \times (T2))) \times (STS3) \\ & \times (SP1) \times (1 - (1 - JB1) \times (1 - JB2)) \times (RPDU1) \\ & \times ((RPDU2) \times (RPDU3)) \times (RPDU4) \times (RPDU5) \end{aligned} \quad (4.2)$$

Função de Disponibilidade:

$$\begin{aligned} A = P\{\Phi(X) = 1\} = E\{ & (1 - (1 - ACS1) \times (1 - ACS2)) \times (STS1) \times (LVP1) \\ & \times (1 - (1 - (UPS1) \times (T1)) \times (1 - (UPS2) \times (T2))) \times (STS3) \\ & \times (SP1) \times (1 - (1 - JB1) \times (1 - JB2)) \times (RPDU1) \\ & \times ((RPDU2) \times (RPDU3)) \times (RPDU4) \times (RPDU5)\} \end{aligned} \quad (4.3)$$

A Tabela 4.2 exibe as variáveis usadas nas funções mencionadas: 4.1, 4.2, 4.3, e seus respectivos equipamentos. Para mais detalhes sobre a notação das variáveis e procedimentos, o leitor é redirecionado a [46].

Tabela 4.2: Equipamentos e variáveis associadas - Estudo 1.

Equipamento	Variáveis
AC Source 1 and 2	AC S1 and AC S2
STS 1 and 3	STS1 and STS3
Sub Panel 1	SP1
Low Voltage Panel 1	LVP1
UPS 1 and 2	UPS1 and UPS2
Junction Box 1 and 2	JB1 and JB2
Power Strip	RPDU(1..5)

Embora nas três equações as variáveis possuam os mesmos nomes, seus valores diferem de acordo com a equação utilizada. Para a Equação 4.1 os valores atribuídos as variáveis são lógicos (verdadeiro ou falso) assim como seu resultado. No caso da Equação 4.2 os valores das variáveis e da função estrutural podem ser 0 ou 1. Já para a Equação 4.3 as variáveis e o resultado da equação de disponibilidade podem ser valores entre 0 e 1.

A Tabela 4.3 apresenta uma relação entre as equações lógica, estrutural, de disponibilidade e os valores cabíveis em cada uma (verdadeiro ou falso, 0 ou 1, fração entre 0 e 1), além de todas as variáveis utilizadas pelas equações.

Tabela 4.3: Relação entre equações, valores e variáveis.

Equação	Valor	Variáveis
Lógica (4.1)	verdadeiro ou falso	AC S1, AC S2, STS1, STS3,
Estrutural (4.2)	0 ou 1	SP1, LVP1, UPS1, UPS2),
Disponibilidade (4.3)	entre 0 e 1	JB1, JB2, RPDU(1..5)

Conhecido o valor da disponibilidade para determinada arquitetura, um modelo EFM é criado, para se computar a exergia, o custo, a energia, e a eficiência do sistema. Para cada uma das sete arquiteturas avaliadas um modelo EFM é definido.

É importante ressaltar que os Racks não são representados em série no modelo EFM, pois cada um dos cinco racks se conecta com os *Junction Box*. Logo, uma falha em algum *Junction Box* não implica em falha do sistema, uma vez que há outro caminho para o fluxo de energia (desde que esse tenha capacidade para suportar a demanda). Note que o EFM representa o fluxo elétrico demandado da arquitetura, diferentemente do RBD. A Figura 4.5 exibe um modelo EFM para arquitetura A5.

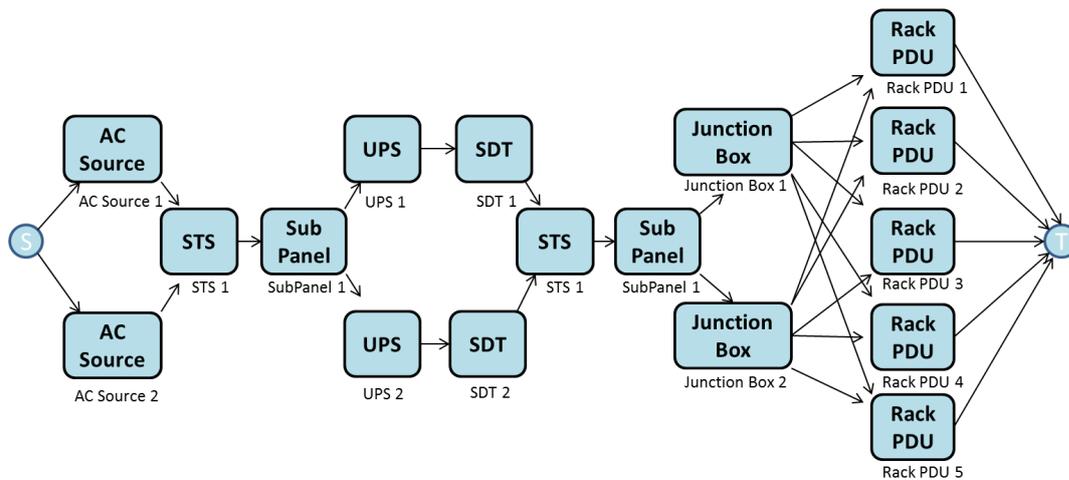


Figura 4.5: Modelo EFM para Arquitetura A5

4.1.3 Resultados

A Tabela 4.4 resume os resultados de cada infraestrutura elétrica, onde A1 a A7 representam as arquiteturas avaliadas. A linha "Conf. Padrão" fornece os resultados obtidos através da configuração padrão da ferramenta Mercury; a linha "Com ADCE" reflete os resultados com a execução do ADCE; *Melhora (%)* é a melhoria alcançada em níveis de percentagem; *Exergia* é a exergia operacional em gigajoule (GJ); *Custo* é o custo

4.1. ESTUDO DE CASO 1 - CENTRO DE DADOS

operacional em USD (dólar); *Disp* é o nível de disponibilidade da arquitetura; *Núm de 9s* é a disponibilidade em números de 9s; *Downtime* é o número de horas inativas por ano; *Energia* é a energia necessária ao funcionamento do sistema em kW/h; e *Eficiência* é a eficiência do sistema.

Tabela 4.4: Resultados da execução do ADCE em centro de dados com melhora em %

Arquiteturas	-----	Exergia	Custo (\$)	Disp	Núm de 9s	Downtime (h)	Energia	Eficiência (%)
A1	Conf. Padrão	5627	1133569	0.9979	2.684	18.121	1178.82	84.82
	Com ADCE	5627	1133569	0.9979	2.684	18.121	1178.82	84.82
	Melhora (%)	0	0	0	0	0	0	0
A2	Conf. Padrão	6922	1174589	0.9994	3.255	4.8667	1219.63	81.99
	Com ADCE	5823	1140993	0.9994	3.255	4.8667	1184.75	84.4
	Melhora (%)	18.87	2.94	0	0	0	2.94	2.94
A3	Conf. Padrão	8253	1215242	0.99943	3.25	4.92	1261	79.24
	Com ADCE	6010	1146719	0.99943	3.25	4.92	1190	83.98
	Melhora (%)	37.32	5.98	0	0	0	5.97	5.98
A4	Conf. Padrão	9005	1238130	0.9993	3.171	5.902	1285	77.77
	Com ADCE	6010	1146591	0.9993	3.171	5.902	1190	83.98
	Melhora (%)	49.83	7.98	0	0	0	7.98	7.99
A5	Conf. Padrão	9398	1250134	0.9993	3.172	5.8887	1298	77.02
	Com ADCE	6010	1146592	0.9993	3.172	5.8887	1190	83.98
	Melhora (%)	56,37	9,03	0	0	0	9,08	9,04
A6	Conf. Padrão	10383	1280685	0.9997	3.698	1.7547	1329	75.22
	Com ADCE	5825	1141398	0.9997	3.698	1.7547	1184	84.4
	Melhora (%)	78.25	12.2	0	0	0	12.25	12.2
A7	Conf. Padrão	11410	1312259	0.9999	5.11	0.067	1361.84	73.43
	Com ADCE	5639	1135910	0.9999	5.11	0.067	1178	84.83
	Melhora (%)	102,34	15.52	0	0	0	15.61	15.52

A partir da Tabela 4.4, deve-se notar a melhoria obtida em várias métricas, destacar, inclusive, que em seis das sete arquiteturas avaliadas os resultados foram positivos. Na arquitetura A2 a exergia foi reduzida em 18,87% o que representa uma redução considerável do impacto ambiental, juntamente com a energia necessária ao funcionamento do ambiente (2,94%) e o custo operacional (2,94%), enquanto a eficiência do sistema melhorou (2,94%). Nas demais arquiteturas (A3 a A7), houve melhora nas mesmas métricas destacadas, mas com diferentes valores para cada arquitetura.

A arquitetura A7 merece destaque, uma vez que apresenta consideráveis índices de melhora, como a redução do impacto ambiental em mais da metade do valor original, de 11.414 para 5.639, ou seja, 102,34%. O custo operacional do centro de dados alcançou uma redução de 15,52%, valor considerável em qualquer empresa, uma vez

4.1. ESTUDO DE CASO 1 - CENTRO DE DADOS

que os recursos financeiros são limitados, além da redução da energia necessária ao funcionamento do ambiente de 15,61% e da melhora da eficiência do sistema em 15,52%.

A arquitetura A1 foi a única na qual não foi observada melhora, comportamento já esperado, uma vez que há ausência de caminhos alternativos para o fluxo elétrico.

É válido acrescentar que o ADCE não afeta as propriedades de dependabilidade, por este motivo não há alteração no resultado das métricas: disponibilidade, números de 9 e indisponibilidade do sistema.

Uma vez adotados maiores níveis de redundância (considerando-se dispositivos com diferentes eficiências energéticas), melhores resultados foram obtidos com o algoritmo ADCE. Esse fato é facilmente percebido por intermédio de uma representação gráfica dos resultados apresentados na Tabela 4.4. A Figura 4.6 apresenta este gráfico, onde o eixo das ordenadas, em zero, representa a avaliação das arquiteturas para as métricas de exergia, custo, energia necessária e eficiência do sistema sem aplicação do algoritmo ADCE. Os valores exibidos acima do ponto zero, nas ordenadas, representam a melhoria das métricas destacadas após a aplicação do algoritmo ADCE nas respectivas arquiteturas.

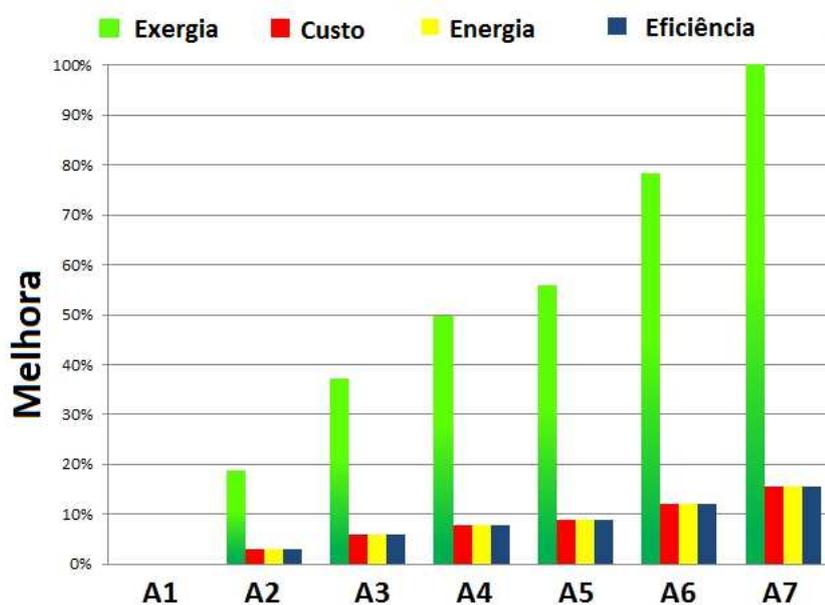


Figura 4.6: Melhora das arquiteturas com a aplicação do ADCE

4.2 Estudo de Caso 2 - Computação em Nuvem

Neste segundo estudo de caso, é feita uma avaliação da infraestrutura elétrica de uma nuvem privada (*private cloud*). Seis arquiteturas de infraestruturas elétricas de uma nuvem privada foram avaliadas. É importante destacar que diferentemente do estudo de caso da seção 4.1, neste foi feita uma avaliação de carga da infraestrutura de TI da nuvem. A demanda elétrica necessária a nuvem em tela foi de 550kw, obtidos por intermédio da ferramenta *HP Power Advisor* [42], desconsiderando a refrigeração do ambiente.

As seguintes métricas foram calculadas para cada arquitetura: (i) exergia operacional, (ii) custos operacionais (iii) disponibilidade, (iv) tempo de inatividade em horas (*downtime*), (v) potência de entrada, e (vi) eficiência do sistema. O período considerado nos cálculos foi de um ano (ou 8760 horas), e os resultados comparados com a configuração padrão da ferramenta Mercury e com a aplicação do algoritmo de distribuição de carga elétrica (ADCE). As Figuras 4.7 a 4.9 exibem as arquiteturas elétricas da nuvem privada avaliada neste estudo.

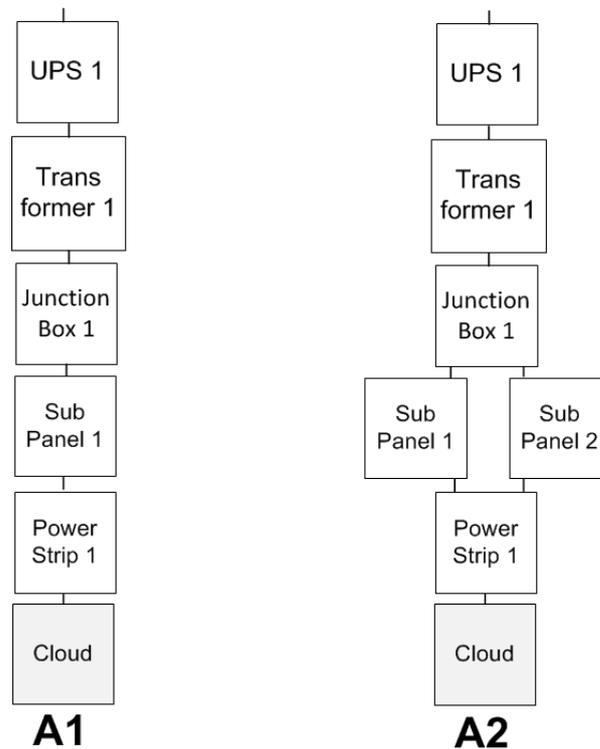


Figura 4.7: Arquiteturas elétricas de nuvens privadas A1 e A2.

4.2. ESTUDO DE CASO 2 - COMPUTAÇÃO EM NUVEM

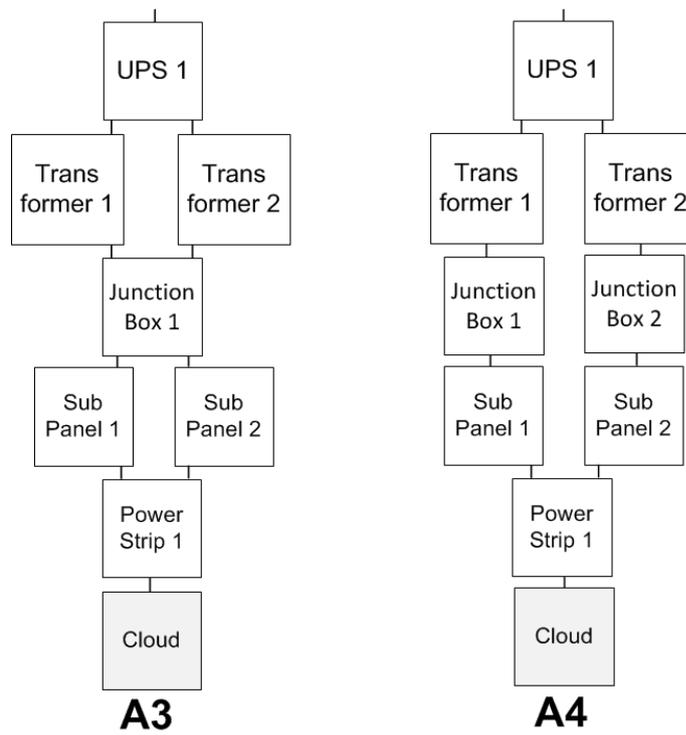


Figura 4.8: Arquiteturas elétricas de nuvens privadas A3 e A4.

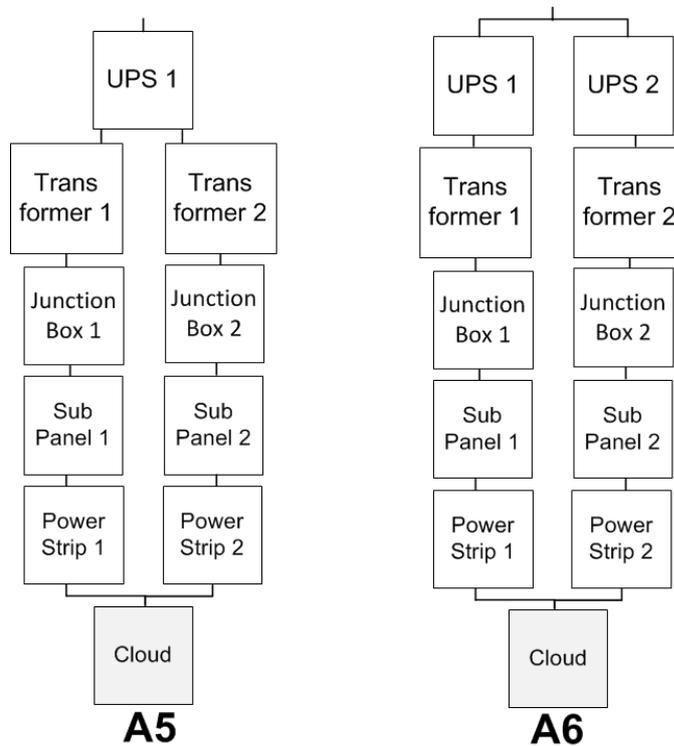


Figura 4.9: Arquiteturas elétricas de nuvens privadas A5 e A6.

4.2.1 Arquiteturas

Neste estudo de caso são avaliadas seis arquiteturas elétricas de uma nuvem privada. A infraestrutura elétrica base para o estudo esta representada pela Figura 4.7(A1), que é composta por um *UPS*, um *Transformer*, um *JunctionBox*, um *SubPanel*, um *PowerStrip*, e pela nuvem, sendo esta a infraestrutura básica adotada neste segundo estudo.

As demais arquiteturas foram criadas a partir do índice de importância de confiabilidade e custo (RCI), e posteriormente classificadas com a nomenclatura: A2, A3, A4, A5 e A6. O RCI indica os componentes a serem replicados. É importante destacar que a infraestrutura de TI da nuvem foi mantida constante, replicando-se apenas os componentes elétricos. Ademais, deve-se ressaltar que o fluxo elétrico em uma nuvem privada é semelhante ao descrito para o estudo de caso da seção 4.1.

O quadrado cinza, que está presente em todas as arquiteturas descritas nas Figuras 4.7, 4.8 e 4.9, representam a infraestrutura de TI da nuvem. A Figura 4.10 apresenta esta infraestrutura, que é composta por cinco servidores e dois *switches*. Esses servidores estão configurados com os sistemas de controlador da nuvem (CC), controlador de *cluster* (CLC) e controladores de nós (NCs). Este modelo de arquitetura foi baseado em uma nuvem privada real, do grupo de estudos MoDCS do Centro de Informática da UFPE. O sistema adotado para configurar a nuvem foi o Eucalyptus.

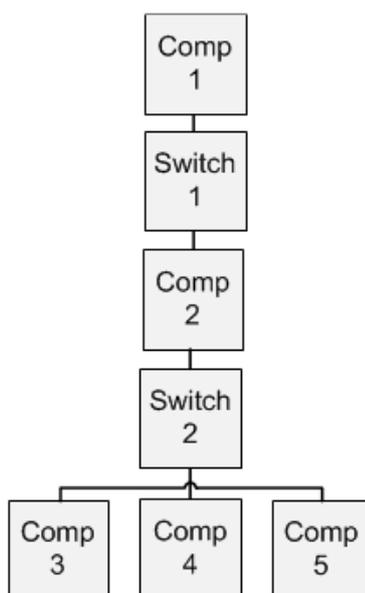


Figura 4.10: Arquitetura de nuvem privada

4.2. ESTUDO DE CASO 2 - COMPUTAÇÃO EM NUVEM

Os computadores que compõem a arquitetura de TI da nuvem possuem características idênticas: mesmo fabricante, modelo, configuração, e custo, como apresentado na Tabela 4.5.

Tabela 4.5: Descrição dos computadores utilizados na Nuvem

Marca/Modelo	Componentes	Descrição
DELL/Power Edge	HD	1 TB
	Memória	8 GB
	CPU	Intel Xeon - 2.2GHZ
Custo Total (USD)		1339,00

Neste estudo de caso os dispositivos elétricos replicados possuem diferentes características (por exemplo, eficiência elétrica e capacidade de potência). A Tabela 4.6 apresenta os valores adotados para a capacidade máxima de potência de cada equipamento, bem como sua eficiência elétrica.

Tabela 4.6: Capacidade e Eficiência

Equipamento	Eficiência(%)	Capacidade(kW)
UPS 1	95.3	5000
UPS 2	90	5000
SDT (or Transformer) 1	98.5	5000
SDT (or Transformer) 2	95	5000
Junction Box 1	99.9	1500
Junction Box 2	98	1500
Sub Panel 1	99.9	1500
Sub Panel 2	95	1500
Power Strip 1	99.5	5000
Power Strip 2	98	5000

4.2.2 Modelos

Esta seção apresenta a modelagem para arquitetura A4, que equivale a infraestrutura elétrica de base A1 com a redundância dos componentes *Transformer*, *Junction Box* e *SubPanel*. A Tabela 4.7 exhibe os tempos de MTTFs e MTTRs adotados para os dispositivos. Estes tempos foram obtidos a partir de [5] [55] [43].

A Figura 4.11 representa o modelo RBD correspondente a arquitetura A4. Para computar os resultados, uma abordagem hierárquica foi adotada. Inicialmente, a infraestrutura de TI (nuvem) foi avaliada e, em seguida, a infraestrutura elétrica. Para obter o resultado da dependabilidade do sistema (TI e elétrica) outro modelo RBD foi criado

Tabela 4.7: Valores de MTTF e MTTR

Equipamento	MTTF	MTTR
Computer	35040	8
Switch	26280	8
ACSource	4380	8
STS	48076	8
Panel	304000	8
UPS	50000	8
SDT	282581	8
JunctionBox	5224000	8
PowerStrip	230223512	8

com ambos. Estes dois blocos foram dispostos em série e preenchidos com o resultado da dependabilidade de cada subsistema. Uma vez que esse modelo RBD é avaliado (com as duas infraestruturas), a dependabilidade de todo o sistema é obtida. Nesse contexto, o bloco correspondente a *cloud* na Figura 4.11, possui os valores dos cálculos feitos sobre a arquitetura de TI, semelhante a Figura 4.10 e compactados em um único bloco.

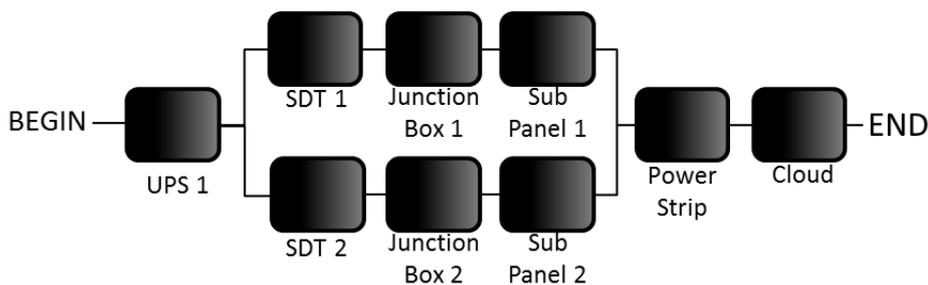


Figura 4.11: Arquitetura A4 na modelagem RBD.

Embora apenas a arquitetura A4 esteja modelada, um processo semelhante foi seguido para avaliar as demais arquiteturas (A1, A2, A3, A5 e A6). Como é possível perceber, há um alto grau de redundância nesta arquitetura, onde a falha dos componentes (*SDT1*, *Junction Box1* e *SubPanel 1*) ou (*SDT2*, *Junction Box2* e *SubPanel 2*) não implicam em uma falha do sistema. Em outras palavras, a nuvem continua a receber a energia necessária (demanda) a seu funcionamento. A falha só ocorrerá se (*SDT1* e *SDT2*) ou (*Junction Box1* e *Junction Box2*) ou (*SubPanel 1* e *SubPanel 2*) falharem.

Logo em seguida, segue um exemplo de funções derivadas da modelagem RBD da arquitetura A4, apresentada na Figura 4.11.

Função Lógica:

$$\Psi(S) = (UPS1) \wedge (((SDT1) \wedge (JB1) \wedge (SP1)) \vee ((SDT2) \wedge (JB2) \wedge (SP2))) \wedge (PS) \wedge (C) \quad (4.4)$$

Função Estrutural:

$$\Phi(X) = (UPS) \times (1 - (1 - (SDT1) \times (JB1) \times (SP1)) \times (1 - (SDT2) \times (JB2) \times (SP2))) \times (PS) \times (C) \quad (4.5)$$

Função de Disponibilidade:

$$A = P\{\Phi(X) = 1\} = E\{(UPS) \times (1 - (1 - (SDT1) \times (JB1) \times (SP1)) \times (1 - (SDT2) \times (JB2) \times (SP2))) \times (PS) \times (C)\} \quad (4.6)$$

Assim como na seção 4.1, os mesmos nomes de variáveis são utilizadas nas equações. A Tabela 4.8 apresenta a relação entre as equações e seus devidos valores, bem como as variáveis utilizadas.

Tabela 4.8: Relação entre equações, valores e variáveis - Estudo 2.

Equação	Valor	Variáveis
Lógica (4.4)	verdadeiro ou falso	UPS, SDT1, JB1,
Estrutural (4.5)	0 ou 1	SP1, SDT2, JB2,
Disponibilidade (4.6)	entre 0 e 1	SP2, PS, C

Com o modelo RBD da arquitetura A4, as métricas de dependabilidade são obtidas, assim, deve-se criar um modelo equivalente a arquitetura A4 no EFM. Nesse modelo EFM é possível calcular as métricas relevantes a este estudo, como exergia operacional, potência de entrada, custo, *downtime* e eficiência do sistema, além das métricas de dependabilidade.

A Figura 4.12 representa o modelo EFM para a arquitetura A4. É importante perceber que a *cloud* foi omitida na modelagem EFM, uma vez que não se trata de equipamento de infraestrutura elétrica e sim de TI, entretanto o valor demandado para o funcionamento da mesma (550kW) foi adicionado ao nó *TargetPoint1* (valor inserido no momento da criação do modelo).

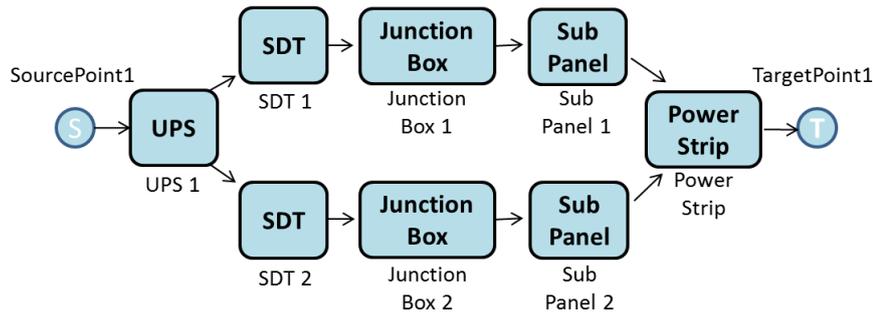


Figura 4.12: Arquitetura A4 na modelagem EFM.

4.2.3 Resultados

A Tabela 4.9 apresenta os resultados separadamente para cada infraestrutura elétrica da nuvem, onde as arquiteturas A1 a A6 representam as arquiteturas avaliadas. Nesta tabela, a linha "Conf. Padrão" representa o resultado obtido com a configuração padrão da ferramenta Mercury, onde o fluxo do modelo é dividido de forma proporcional a quantidade de arestas ascendentes a cada vértice, a linha "Com ADCE" é o resultado obtido com a execução do ADCE; *Melhora (%)* é a melhoria alcançada em níveis de percentagem.

Tabela 4.9: Resultados da execução do ADCE em nuvens com melhora em %.

Arquiteturas		Exergia (GJ)	Custo (\$)	Disp (9's)	Downtime (h)	Energia (kW)	Eficiência(%)
A1	Conf. Padrão	1262	568436	3.66	1.89	590	93
	Com ADCE	1262	568436	3.66	1.89	590	93
	Melhora (%)	0	0	0	0	0	0
A2	Conf. Padrão	1742	583111	3.72	1.66	605	90
	Com ADCE	1262	568451	3.72	1.66	590	93.4
	Melhora (%)	38	2.5	0	0	2.5	2.5
A3	Conf. Padrão	2093	593870	3.78	1.42	616	89
	Com ADCE	1262	568467	3.78	1.42	590	93
	Melhora (%)	65	4.5	0	0	4.5	4.5
A4	Conf. Padrão	2299	600149	3.7	1.4	620	88
	Com ADCE	1262	568468	3.7	1.4	590	93
	Melhora (%)	82	5.6	0	0	5.4	5.7
A5	Conf. Padrão	2457	604985	3.79	1.4	627	87
	Com ADCE	1262	568469	3.79	1.4	590	93
	Melhora (%)	94	6.4	0	0	6.3	6.9
A6	Conf. Padrão	3076	623973	7.56	0.0002	647	84
	Com ADCE	1262	568559	7.56	0.0002	590	93
	Melhora (%)	143	9.7	0	0	9.7	10.7

4.2. ESTUDO DE CASO 2 - COMPUTAÇÃO EM NUVEM

Ainda na Tabela 4.9, a coluna *Exergia* representa a Exergia Operacional em gigajoules (GJ), *Custo* é o custo operacional em dólares (USD), *Disp (9's)* expressa a disponibilidade em número de 9s da arquitetura; *Downtime* é o número de horas de inatividade por ano. *Energia* é a energia necessária ao funcionamento do sistema em kW/h; e *Eficiência* é a eficiência do sistema. O período considerado para avaliação destas métricas foi de um ano, ou 8760 horas.

Com a comparação das métricas obtidas com a configuração padrão do Mercury e com a aplicação do ADCE, é possível perceber que houve melhora nas arquiteturas A2, A3, A4, A5 e A6. A redução do consumo elétrico das arquiteturas ficou entre 2,5% a 9,7%. Se há uma diminuição no consumo elétrico, haverá também uma diminuição do impacto ambiental e principalmente uma redução dos custos (fator decisivo para tomada de decisão das empresas).

Em destaque, o impacto ambiental chegou ao expressivo valor de 143% de redução para a arquitetura A6, reduzindo a exergia de 3.076 para 1.262, e os custos de operação da nuvem em 9,7%, chegando a uma melhora da eficiência do sistema de 10,7%.

A Figura 4.13 exibe uma representação gráfica dos resultados da aplicação do ADCE nas seis arquiteturas de nuvem privada avaliadas. Apenas as métricas que sofreram alteração de seu valor após a aplicação do algoritmo foram representadas.

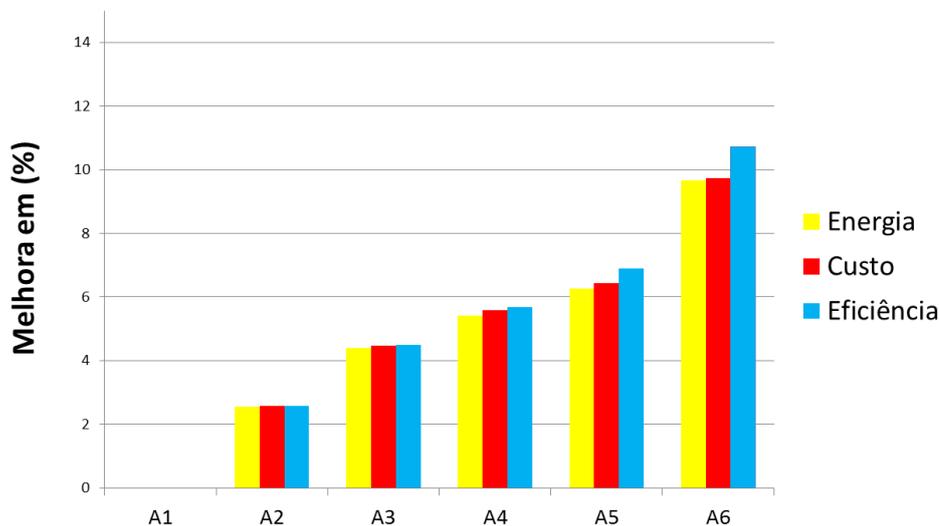


Figura 4.13: Melhora das métricas após aplicação do ADCE

Para destacar os valores do consumo de energia, do custo e da eficiência do sistema, a métrica da sustentabilidade foi omitida deste gráfico e a percentagem de melhora

apresentada entre 0 e 15% (eixo das ordenadas). A arquitetura A1 é a única que não apresenta melhora, pois não há outro fluxo possível para a eletricidade, logo o ADCE não tem como melhorar essa distribuição.

4.3 Considerações Finais

Este capítulo apresentou estudos de caso para verificar a aplicabilidade do ADCE em infraestruturas elétricas. Seis infraestruturas elétricas de um centro de dados foram analisadas, baseadas em uma arquitetura real (HP Labs, Palo Alto, EUA). Em seguida, outro estudo foi feito no qual foram analisadas cinco arquiteturas de uma nuvem privada real do grupo de pesquisa MoDCS do centro de informática da UFPE.

5

Conclusão

Com o aumento da dependência dos serviços de Internet nos últimos anos, acompanhado dos novos paradigmas da computação (por exemplo, computação em nuvem), a disponibilidade em centros de dados tornou-se uma preocupação de alta prioridade. Assim, para empresas que possuem grande dependência de suas operações ligadas ao funcionamento da Internet, interrupções de serviço podem ser muito caras, atingindo facilmente os milhões de dólares por hora.

Um princípio de design amplamente utilizado na tolerância a falhas é a introdução de redundância para melhorar a disponibilidade dos ambientes. No entanto, o acréscimo de redundância pode elevar o consumo de energia, incorrendo em um impacto negativo na sustentabilidade e no custo das operações.

Entretanto, em paralelo aos altos índices de disponibilidade almejados pelas empresas e seus usuários, há uma preocupação com as questões de sustentabilidade. É importante que as empresas demonstrem seu interesse em possuir alta disponibilidade aliada a estratégias de diminuição do impacto ambiental.

Em centros de dados o impacto ambiental é medido pelo consumo de energia. Quanto maior o centro de dados, mais energia será consumida e maiores podem ser os índices que afetam negativamente a sustentabilidade (emissão de gás carbônico, exergia, etc.). De acordo com a matéria prima geradora da eletricidade, haverá uma variação do impacto ambiental, por exemplo, o impacto através da utilização do carvão mineral, será maior que através de hidroelétricas.

Diante desse problema, existe um modelo chamado EFM, que avalia de forma integrada a dependabilidade, o custo e as questões de sustentabilidade de infraestruturas elétricas. Esse modelo está implementado em uma ferramenta de nome Mercury, que possui ambientes habilitados a elaboração de modelos formais como RBD, SPN e MC. No entanto, na modelagem EFM a distribuição padrão do fluxo elétrico é feita de forma

proporcional a quantidade de vértices ascendentes que cada aresta possui no modelo, ou a distribuição fica sob a responsabilidade do projetista do ambiente, sujeito a erros humanos em sua elaboração.

Assim, essa dissertação propôs um algoritmo de distribuição de carga elétrica (ADCE), visando preencher a lacuna desta modelagem, fornecendo uma boa heurística, reduzindo o consumo de energia (quando possível) de modelos de infraestrutura elétrica projetados na ferramenta Mercury, utilizando o modelo EFM. A metodologia adotada consiste em uma implementação do algoritmo proposto na visão de EFM da ferramenta Mercury, possibilitando a melhora da distribuição do fluxo elétrico nos modelos EFM para infraestruturas elétricas.

Com o objetivo de verificar a aplicabilidade do algoritmo proposto, dois estudos de caso foram desenvolvidos, onde foram analisadas 13 infraestruturas elétricas. Estas arquiteturas foram baseadas em infraestruturas reais de um centro de dados da HP, localizado em Palo Alto, nos EUA e de uma nuvem privada, localizada no laboratório do grupo de estudos MoDCS do centro de informática da UFPE.

Os resultados da avaliação das métricas das arquiteturas com a aplicação do algoritmo ADCE foram melhores ou iguais aos resultados obtidos usando a configuração padrão da ferramenta Mercury. Para os centros de dados foi alcançada uma redução no consumo de energia entre 2,9% e 15,5% e o impacto ambiental (exergia) reduzido de 11.414 para 5.693, ou seja, 102%. Já para o caso das nuvens privadas a redução do consumo de energia ficou entre 2,5% e 9,7% e da exergia uma redução de 3.076 para 1.262, ou seja, de 143%.

É importante ressaltar que todos os objetivos traçados no início deste trabalho foram alcançados, como:

- Uma heurística foi proposta para diminuir o consumo elétrico dos modelos EFM.
- Esta heurística foi transformada em algoritmo (ADCE) e implementada na ferramenta Mercury.
- O algoritmo foi integrado aos métodos de análise do modelo EFM.
- Duas arquiteturas elétricas foram avaliadas com esta estratégia.

5.1 Contribuições

Como resultado do trabalho apresentado nesta dissertação, as seguintes contribuições podem ser destacadas:

- Benefícios financeiros com a redução do custo de operação dos ambientes de centro de dados ou nuvens.
- Benefícios globais com a redução do impacto ambiental por intermédio da redução do consumo de energia.
- Arquiteturas energeticamente mais eficientes.

Além da contribuição mencionada, alguns artigos que apresentam os resultados desta dissertação foram produzidos:

1. *João Ferreira, Gustavo Callou and Paulo Maciel. A Power Load Distribution Algorithm to Optimize Data Center Electrical Flow. Energies - Open Access Journal. 2013. ISSN: 1996-1073. DOI>10.3390/en6073422*
2. *João Ferreira, Gustavo Callou, Jamilson Dantas, Rafael Souza, and Paulo Maciel. An Algorithm to Optimize Electrical Flows. In: Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2013). Manchester, United Kingdom.*

Entretanto, os resultados colhidos com o curso de mestrado não se restringem às publicações alcançadas apenas com esta proposta, outros resultados foram também obtidos por intermédio de parcerias com os integrantes do grupo de estudos MoDCS, que até o momento da publicação desta dissertação foram:

1. *Callou, Gustavo ; Maciel, Paulo ; Tutsch, Dietmar ; Arajo, Julian ; Ferreira, João ; Souza, Rafael . A Petri Net-Based Approach to the Quantification of Data Center Dependability. In: Pawel Pawlewski. (Org.). Petri Nets - Manufacturing and Computer Science. 1ed.: InTech, 2012, v. , p. 313-336.*
2. *Callou, Gustavo ; Maciel, Paulo ; Tutsch, Dietmar ; Ferreira, João ; Araújo Julian ; Souza, Rafael. Estimating sustainability impact of high dependable data centers: a comparative study between Brazilian and US energy mixes. Computing (Wien. Print), v. v, p. 1-34, 2013.*

3. *Rafael Souza, Gustavo Callou, Kadna Camboim, João Ferreira, and Paulo Maciel. The Effects of Temperature Variation on Data Center IT Systems. In: Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2013). Manchester, United Kingdom.*

5.2 Trabalhos Futuros

Embora esta dissertação proponha um algoritmo para redução do consumo elétrico dos modelos EFM, que considera questões relativas à sustentabilidade, confiabilidade e custo para arquiteturas elétricas, há outras possibilidades para se melhorar e ampliar o trabalho atual. Os itens a seguir resumem algumas possibilidades:

- **Avaliação dinâmica** - Atualmente os modelos EFM oferecem a seus usuários uma visualização estática da arquitetura, acredita-se que com um gerenciamento dinâmico das arquiteturas elétricas os projetistas possam obter ganhos significativos nas avaliações.
- **Generalização do ADCE** - Embora o algoritmo de distribuição de carga elétrica (ADCE) tenha sido proposto para um fim específico, é possível torna-lo genérico, uma vez executado em grafos acíclicos dirigidos, sendo uma nova solução para problemas de várias natureza, como por exemplo, fluxo máximo ou mínimo.
- **Considerar infraestruturas de resfriamento** - Como esta dissertação considerou apenas os componentes da infraestrutura elétrica, seria interessante considerar a infraestrutura de resfriamento, responsável por 50% do consumo de energia dos centros de dados ou nuvens.
- **Integração com aplicações embarcadas** - De posse dos resultados do ADCE no modelo EFM, seria interessante uma integração com os aplicativos embarcados que realizam a gestão elétrica no ambiente projetado.

5.3 Considerações Finais

Este trabalho propôs um algoritmo para distribuição da carga elétrica (ADCE) nos modelos EFM, respeitando os limites de capacidade dos componentes. Dois estudos de caso foram elaborados, considerando questões de dependabilidade, sustentabilidade e

custo para validar a aplicabilidade do ADCE. Embora os resultados alcançados tenham sido satisfatórios, há outras possibilidades em aberto para aproveitamento do algoritmo proposto.

Referências

- [1] Abbasi, Z., Varsamopoulos, G., and Gupta, S. (2010). Thermal aware server provisioning and workload distribution for internet data centers. In *ACM International Symposium on High Performance Distributed Computing (HPDC10)*, pages 130–141.
- [2] Administration-EIA., U. E. I. (2011). Sustainability review. *Technical report*.
- [3] Agency, E. P. (2007). Report to congress on server and data center energy efficiency. http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf.
- [4] Al-Qawasmeh, A. (2012). *Heterogeneous Computing Environment Characterization and Thermal-Aware Scheduling Strategies to Optimize data center power consumption*. Ph.D. thesis, Colorado State University.
- [5] Ang, C.-W. and Tham, C.-K. (2007). Analysis and optimization of service availability in a ha cluster with load-dependent machine availability. *Parallel and Distributed Systems, IEEE Transactions on*, **18**(9), 1307–1319.
- [6] Arregoces, M. and Portolani, M. (2003). *Data center fundamentals*. Fundamentals Series. Cisco.
- [7] Augarsten, S. (1984). *Bit by bit: An illustrated history of computers*. Houghton Mifflin Co.
- [8] Avizienis, A., Laprie, J., and Randell, B. (2001). Fundamental Concepts of Dependability. *Technical Report Series-University of Newcastle upon Tyne Computing Science*.
- [9] Baliga, J., Ayre, R., Hinton, K., and Tucker, R. (2011). Green cloud computing: Balancing energy in processing, storage, and transport. *Proceedings of the IEEE*, **99**(1), 149–167.
- [10] Barroso, L. A., Dean, J., and Holzle, U. (2003). Web search for a planet: The google cluster architecture. *Micro, Ieee*, **23**(2), 22–28.
- [11] Bondy, J. A. and Murty, U. S. R. (1976). *Graph theory with applications*, volume 290. Macmillan London.
- [12] Bouley, D. (2010). Estimating a data center’s electrical carbon footprint. *white paper*, **66**.

-
- [13] Brundtland, G. H. (1987). World commission on environment and development. *Our common future*.
- [14] Callou, G., Maciel, P., Magnani, F., and Figueiredo, J. (2011a). Estimating sustainability impact, total cost of ownership and dependability metrics on data center infrastructures. In *Sustainable Systems and Technology (ISSST), 2011 IEEE International Symposium on*, pages 1–6.
- [15] Callou, G., Sousa, E., Maciel, and Magnani, F. (2011b). A formal approach to the quantification of sustainability and dependability metrics on data center infrastructures. In *Proceedings of DEVS*, pages 274–281, San Diego, CA, USA. SCS.
- [16] Callou, G., Maciel, P., Tutsch, D., and Araujo, J. (2012a). Models for dependability and sustainability analysis of data center cooling architectures. In *Dependable Systems and Networks (DSN), 2012 IEEE International Conference on*, pages 1–6.
- [17] Callou, G., Maciel, P., Tutsch, D., Araujo, C., Ferreira, J., and Souza, R. (2012b). *Petri Nets - Manufacturing and Computer Science, Chapter 14 - A Petri Net-Based Approach to the Quantification of Data Center Dependability*. InTech.
- [18] Callou, G., Maciel, P., Tutsch, D., Ferreira, J., Araújo, J., and Souza, R. (2013). Estimating sustainability impact of high dependable data centers: a comparative study between brazilian and us energy mixes. *Computing*, pages 1–34.
- [19] Chen, Y., Gmach, D., Hyser, C., Wang, Z., Bash, C., Hoover, C., and Singhal, S. (2010). Integrated management of application performance, power and cooling in data centers. In *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 615–622. IEEE.
- [20] Christofides, N. (1975). *Graph theory: An algorithmic approach (Computer science and applied mathematics)*. Academic Press, Inc.
- [21] Ciardo, G., Blakemore, A., Chimento, P. F., Muppala, J. K., and Trivedi, K. S. (1993). Automated generation and analysis of markov reward models using stochastic reward nets. *IMA Volumes in Mathematics and its Applications*, **48**, 145–145.
- [22] Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2001). *Introduction to algorithms*. MIT press.
-

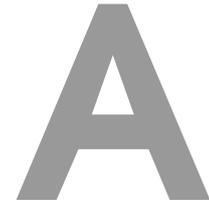
-
- [23] Curtis, A., Carpenter, T., Elsheikh, M., Lopez-Ortiz, A., and Keshav, S. (2012). Rewire: An optimization-based framework for unstructured data center network design. In *INFOCOM, 2012 Proceedings IEEE*, pages 1116–1124.
- [24] Desrochers, A. and Al-Jaar, R. (1995). *Applications of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis*. IEEE Press.
- [25] Dincer, I. and Rosen, M. (2007). *Exergy: energy, environment and sustainable development*. Elsevier Science.
- [26] Distefano, S., Scarpa, M., and Puliafito, A. (2006). Modeling distributed computing system reliability with drbd. In *Reliable Distributed Systems, 2006. SRDS'06. 25th IEEE Symposium on*, pages 106–118. IEEE.
- [27] Ebeling, C. E. (1997). *An introduction to reliability and maintainability engineering*. McGraw Hill New York, NY.
- [28] Eucalyptus (2010). *Cloud Computing and Open Source: IT Climatology is Born*. Eucalyptus Systems, Inc., Goleta, CA.
- [29] Eucalyptus Systems, I. (2012). Eucalyptus - the open source cloud platform.
- [30] Fan, X. and et al (2007). Power provisioning for a warehouse-sized computer.
- [31] Ford, D. R. and Fulkerson, D. R. (2010). *Flows in Networks*. Princeton University Press, Princeton, NJ, USA.
- [32] Gandhi, A. and Harchol-Balter, M. (2011). How data center size impacts the effectiveness of dynamic power management. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 1164–1169. IEEE.
- [33] Ganesh, L., Weatherspoon, H., Marian, T., and Birman, K. (2013). Integrated approach to data center power management.
- [34] Gil Montoya, F., Manzano-Agugliaro, F., Gómez López, J., and Sánchez Alguacil, P. (2012). Power quality research techniques: Advantages and disadvantages. *DYNA*, **79**(173), 66–74.
- [35] Gmach, D., Chen, Y., Shah, A., Rolia, J., Bash, C., Christian, T., and Sharma, R. (2010). Profiling sustainability of data centers. In *Sustainable Systems and Technology (ISSST), 2010 IEEE International Symposium on*, pages 1–6. IEEE.
-

-
- [36] Gong, C., Liu, J., Zhang, Q., Chen, H., and Gong, Z. (2010). The characteristics of cloud computing. In *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on*, pages 275–279. IEEE.
- [37] Gorbenko, A., Kharchenko, V., and Romanovsky, A. (2007). On composing dependable web services using undependable web components. *International Journal of Simulation and Process Modelling*, **3**(1), 45–54.
- [38] Greenberg, A., Hamilton, J., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D., Patel, P., and Sengupta, S. (2009). V12: a scalable and flexible data center network. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 51–62. ACM.
- [39] Guo, C., Wu, H., Tan, K., Shi, L., Zhang, Y., and Lu, S. (2008). Dcell: a scalable and fault-tolerant network structure for data centers. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 75–86. ACM.
- [40] Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., Shi, Y., Tian, C., Zhang, Y., and Lu, S. (2009). Bcube: a high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Computer Communication Review*, **39**(4), 63–74.
- [41] Gyarmati, L. and Trinh, T. (2010). How can architecture help to reduce energy consumption in data center networking? In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, pages 183–186. ACM.
- [42] Hewlett-Packard (2013). Hp power advisor. <http://h18004.www1.hp.com/products/solutions/power/index.html>.
- [43] IEEE423 (1998). *IEEE Gold Book 473, Design of Reliable Industrial and Commercial Power Systems*.
- [44] K.Mukherjee and G.Sahoo (2010). Article:green cloud: An algorithmic approach. *International Journal of Computer Applications*, **9**(9), 1–6. Published By Foundation of Computer Science.
- [45] Kuo, W. and Zuo, M. J. (2003). *Optimal Reliability Modeling - Principles and Applications*. Wiley.
- [46] Maciel, P., Trivedi, K., Matias, R., and Kim, D. (2011). *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*, chapter 3 - Dependability Modeling, pages 53–97. Information Science Reference.
-

-
- [47] Marin, P. S. (2011). *Data centers: desvendando cada passo: conceitos, projeto, infraestrutura física e eficiência energética*. Editora Erica.
- [48] Marwah, M., Maciel, P., Shah, A., Sharma, R., Christian, T., Almeida, V., Araujo, C., Souza, E., Callou, G., Silva, B., *et al.* (2010). Quantifying the sustainability impact of data center availability. *ACM SIGMETRICS Performance Evaluation Review*, **37**(4), 64–68.
- [49] Mell, P. and Grance, T. (2011). The nist definition of cloud computing (draft). *NIST special publication*, **800**, 145.
- [50] Murugesan, S. (2008). Harnessing green it: Principles and practices. *IT professional*, **10**(1), 24–33.
- [51] Parhami, B. (1988). From defects to failures: a view of dependable computing. *ACM SIGARCH Computer Architecture News*, **16**(4), 157–168.
- [52] Patterson, M. K. (2008). The effect of data center temperature on energy efficiency. In *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITherm 2008. 11th Intersociety Conference on*, pages 1167–1174. IEEE.
- [53] Rausand, M. and Høyland, A. (2004). *System reliability theory: models, statistical methods, and applications*, volume 396. John Wiley & Sons.
- [54] Rosen, M. A., Dincer, I., and Kanoglu, M. (2008). Role of exergy in increasing efficiency and sustainability and reducing environmental impact. *Energy Policy*, **36**(1), 128–137.
- [55] Schroeder, B. and Gibson, G. (2007). Understanding failures in petascale computers. In *Journal of Physics: Conference Series*, volume 78, page 012022. IOP Publishing.
- [56] Silva, B., Callou, G., Tavares, E., Maciel, P., Figueiredo, J., Sousa, E., Araujo, C., Magnani, F., and Neves, F. (2012). Astro: An integrated environment for dependability and sustainability evaluation. *Sustainable Computing: Informatics and Systems*.
- [57] Trivedi, K. (2002). *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*. Wiley Interscience Publication, 2 edition.
- [58] Turner IV, W. P., PE, J., Seader, P., and Brill, K. (2006). Tier classification define site infrastructure performance. *Uptime Institute*, page 17.
-

- [59] VAN DER PERRE, D., LATRECHE, A., and HEARNDEN, J. (2009). The green grid energy policy research for data centres.
- [60] Wall, G. and Gong, M. (2001). On exergy and sustainable development-part 1: Conditions and concepts. *Exergy, An International Journal*, **1**(3), 128–145.
- [61] Xi, K., Liu, Y., and Chao, H. (2011). Enabling flow-based routing control in data center networks using probe and ecmp. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 608 –613.
- [62] Xie, M., Poh, K., and Dai, Y. (2004). *Computing system reliability: models and analysis*. Springer.
- [63] Xu, L., Tan, G., Zhang, X., and Zhou, J. (2011). Energy aware cloud application management in private cloud data center. In *Cloud and Service Computing (CSC), 2011 International Conference on*, pages 274–279.
- [64] Yamini, R. (2012). Power management in cloud computing using green algorithm. In *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on*, pages 128–133.
- [65] Zhang, X., Zhao, X., Li, Y., and Zeng, L. (2010). Key technologies for green data center. In *Information Processing (ISIP), 2010 Third International Symposium on*, pages 477–480. IEEE.

Appendices



Ambiente ASTRO/Mercury

O ASTRO é um ambiente que analisa infraestruturas de TI, elétrica e de refrigeração dos centros de dados e das nuvens [56]. O núcleo do ambiente é denominado Mercury, cuja responsabilidade consiste em avaliar os modelos: Redes de Petri Estocástica (SPN), Diagrama de bloco de Confiabilidade (RBD), EFM e Cadeia de Markov (MC). O Mercury suporta as dificuldades encontradas com a modelagem RBD (por exemplo, relação de dependência), por intermédio da utilização do SPN ou MC.

A Figura A.1 apresenta a relação entre o ASTRO e o Mercury. É importante ressaltar que o modelo construído pelas visões de TI, elétrica e de refrigeração, disponíveis no ASTRO, pode ser convertido automaticamente para os modelos suportados pelo Mercury, por exemplo, para avaliação de confiabilidade, custo e impacto ambiental.

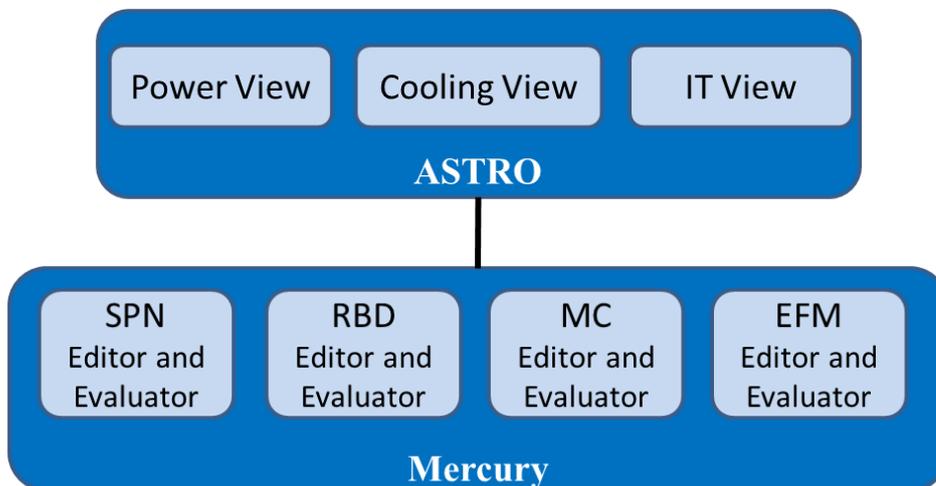


Figura A.1: Funcionalidades do Ambiente ASTRO/Mercury.

Portanto, no ambiente ASTRO os usuários não especializados podem computar a confiabilidade, sustentabilidade e avaliação de custos, sem a necessidade de estarem familiarizados com esses modelos.

A Figura A.2 retrata o ambiente Mercury. O retângulo vermelho foi utilizado para destacar o conjunto de modelos disponíveis na ferramenta (SPN, RBD, CTMC e EFM). Nesta figura, um exemplo da modelagem EFM é também representado, onde se pode perceber claramente um grafo acíclico dirigido.

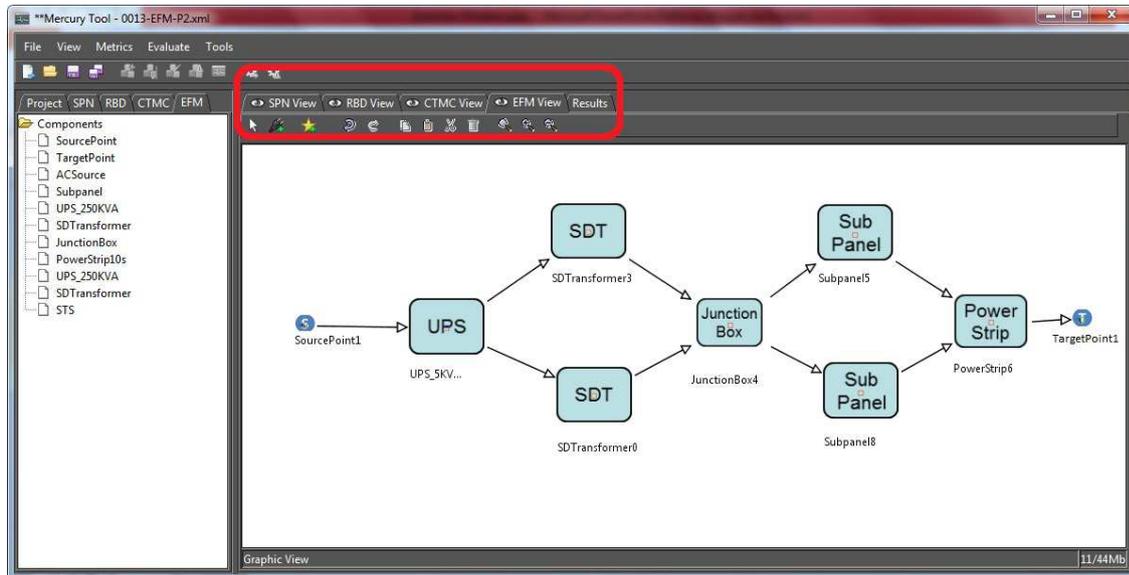


Figura A.2: Exemplo do Ambiente Mercury.

Para maiores detalhes sobre a ferramenta ASTRO/Mercury, o leitor é redirecionado a [56].

Exemplo

A Figura A.3 é um exemplo de uma modelagem em EFM. Os retângulos arredondados equivalem ao tipo do equipamento, e os rótulos aos nomes de cada item. As arestas têm pesos que são usados para direcionar o fluxo de energia entre os componentes. Por uma questão de simplificação na representação gráfica dos modelos EFM, pesos iguais a 1 são omitidos, logo se não há valor visível em uma aresta significa que seu valor é 1 (por exemplo a aresta que conecta *UPS1* ao *STS1*).

Na presente dissertação, o EFM é empregado para calcular a eletricidade total necessária para suprir a demanda informada pelo "*TargetPoint1*". Visando uma melhor compreensão, este exemplo será explicado detalhadamente. Considere que a eletricidade demandada associada ao "*TargetPoint1*" é de 128.25kW. Assim, considerando a eficiência do componente "*STS1*" como de 95%, a energia que o mesmo deve receber como entrada

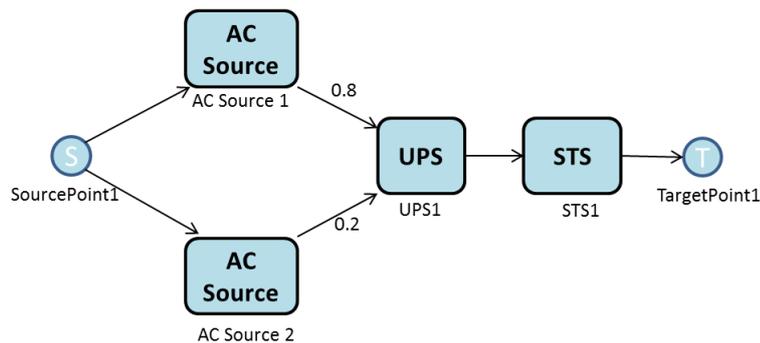


Figura A.3: Exemplo da modelagem EFM.

deve ser de 135kW, para que em sua saída seja gerada 128.25kW.

A mesma estratégia é adotada para o componente "UPS1". Sua eficiência energética é de 90%, logo a energia elétrica que flui através dele é 150kW, para que produza 135kW (necessários ao "STS1"). A partir do "UPS1", o fluxo é dividido entre os dois "AC Sources" de acordo com os pesos das arestas associadas. Portanto, o fluxo de 120kW é atribuído ao "AC Source1" e 30kW ao "AC Source2". O processo continua até atingir o "SourcePoint1", que acumula o fluxo total.

Um peso de 0,8 unidades é atribuído à aresta que liga o nó "UPS1" ao "ACSource1" e um peso de 0,2 para a outra aresta que liga o nó "ACSource2". Isso significa que o "ACSource1" recebe quatro vezes mais fluxo elétrico que "ACSource2".

É importante frisar que os pesos das arestas são definidos pelo projetista de modelo. Logo, não há garantias de que os designers aloquem os melhores valores para a distribuição, incorrendo possivelmente em um maior consumo de energia.