

# Availability Evaluation and Sensitivity Analysis of a Private Cloud with DRBD Models

Danilo Oliveira

Orientador: Paulo Maciel

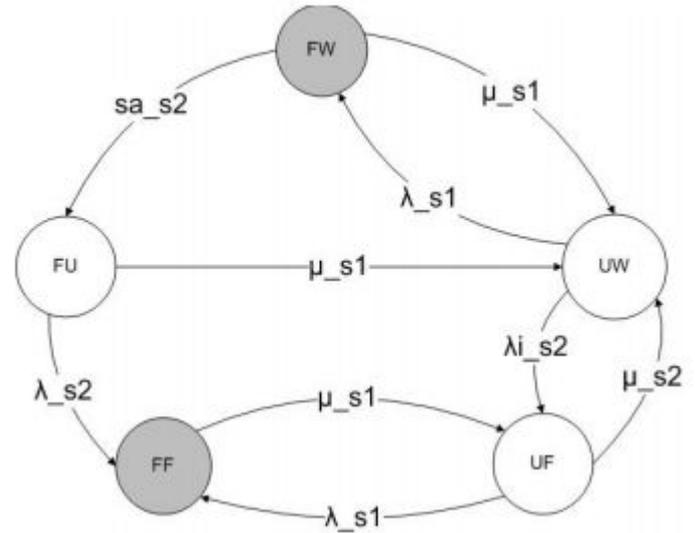
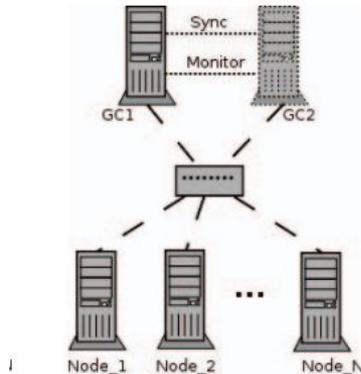
Co-orientador: Nelson Rosa

# Agenda

- Introdução
- Metodologia
- DRBD
- Arquitetura e modelos
- Avaliação de disponibilidade e análise de sensibilidade
- Conclusão e trabalhos futuros

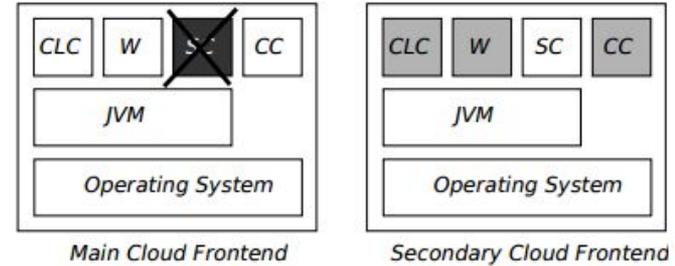
# Introdução

- Neste trabalho, revisitamos a plataforma Eucalyptus. Os modelos que tínhamos até então simplificavam a questão da redundância: a falha de qualquer serviço do frontend primário levaria à ativação do frontend secundário

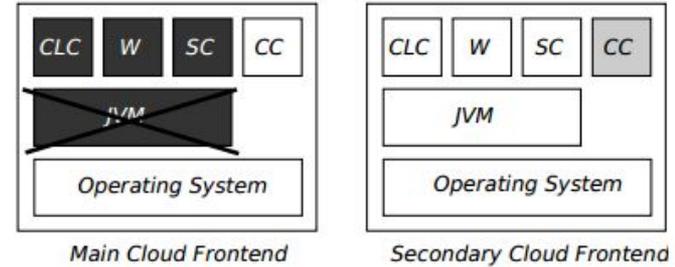


# Introdução

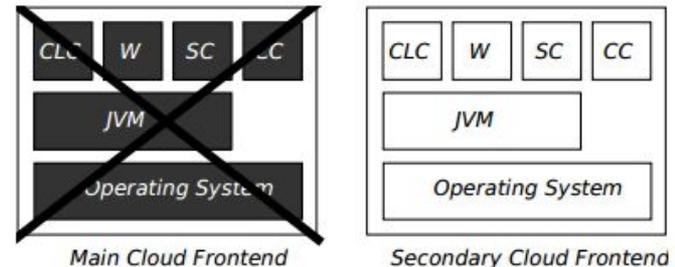
- Entretanto, na implementação do Eucalyptus, a replicação é aplicada entre pares de serviços
- A modelagem de tal cenário é complexa:
  - Impossível de ser representada por RBD tradicionais
  - Explosão do número de estados em cadeias de Markov
  - Rede de Petri com grande número de estruturas repetidas e arcos se cruzando



(a) Failure of an individual component

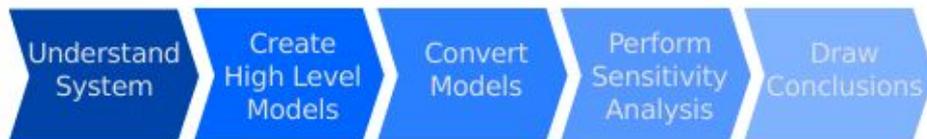


(b) JVM failure



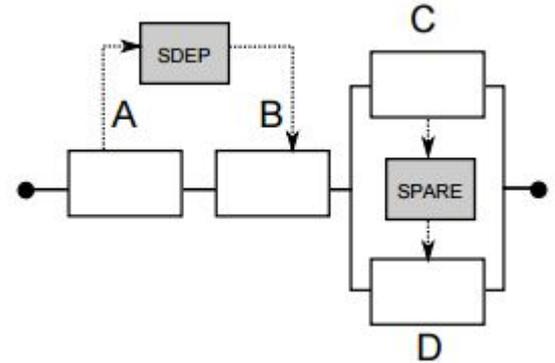
# Introdução

- Neste trabalho, propomos uma metodologia para a modelagem e avaliação de desempenho de plataformas de nuvens privadas, utilizando o Eucalyptus como estudo de caso
- Adotamos o formalismo DRBD (Dynamic Reliability Block Diagram) que oferece a facilidade de modelagem dos RBDs tradicionais, ao mesmo tempo que permite capturar relações de dependências entre componentes
- Os modelos concebidos são submetidos à análise de sensibilidade a fim de obter insights sobre quais componentes provocam maior impacto na disponibilidade do sistema



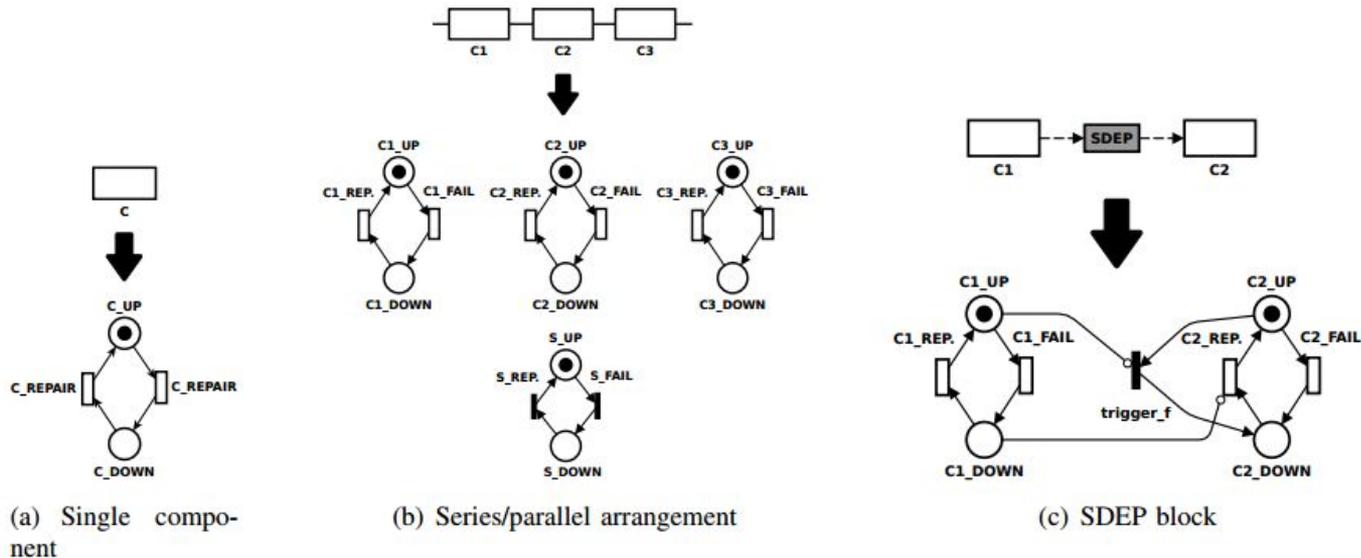
# Dynamic Reliability Block Diagram DRBD

- Extensão do RBD tradicional que permite capturar relações/dependências entre componentes
- O bloco SDEP indica que a falha de um bloco provoca a falha imediata de outros
- Blocos SPARE indicam que dois componentes estão configurados para trabalhar em cold/warm standby

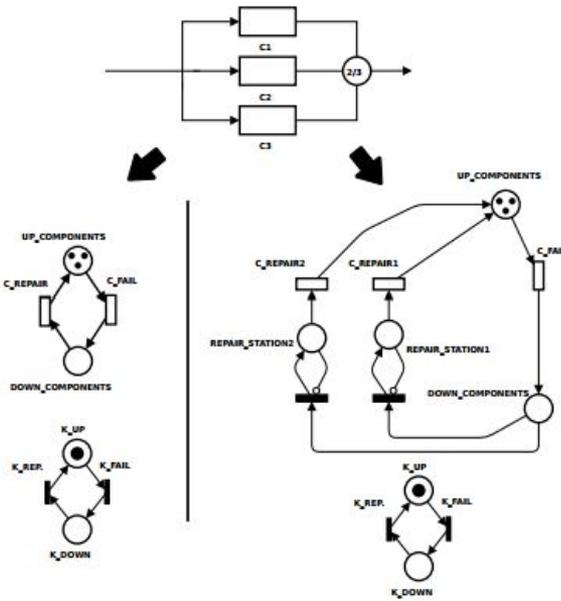


# Dynamic Reliability Block Diagram DRBD

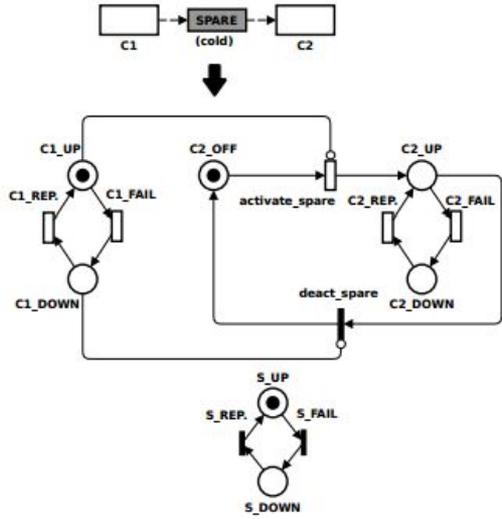
Modelos em DRBD precisam ser convertidos para modelos estocásticos ou de simulação para serem resolvidos



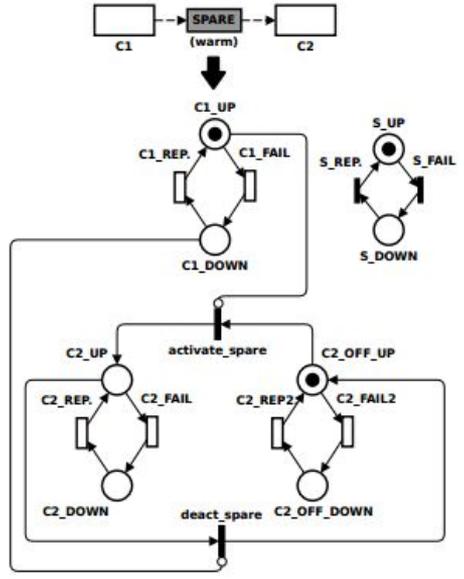
# Dynamic Reliability Block Diagram DRBD



(d) K-out-of-n arrangement

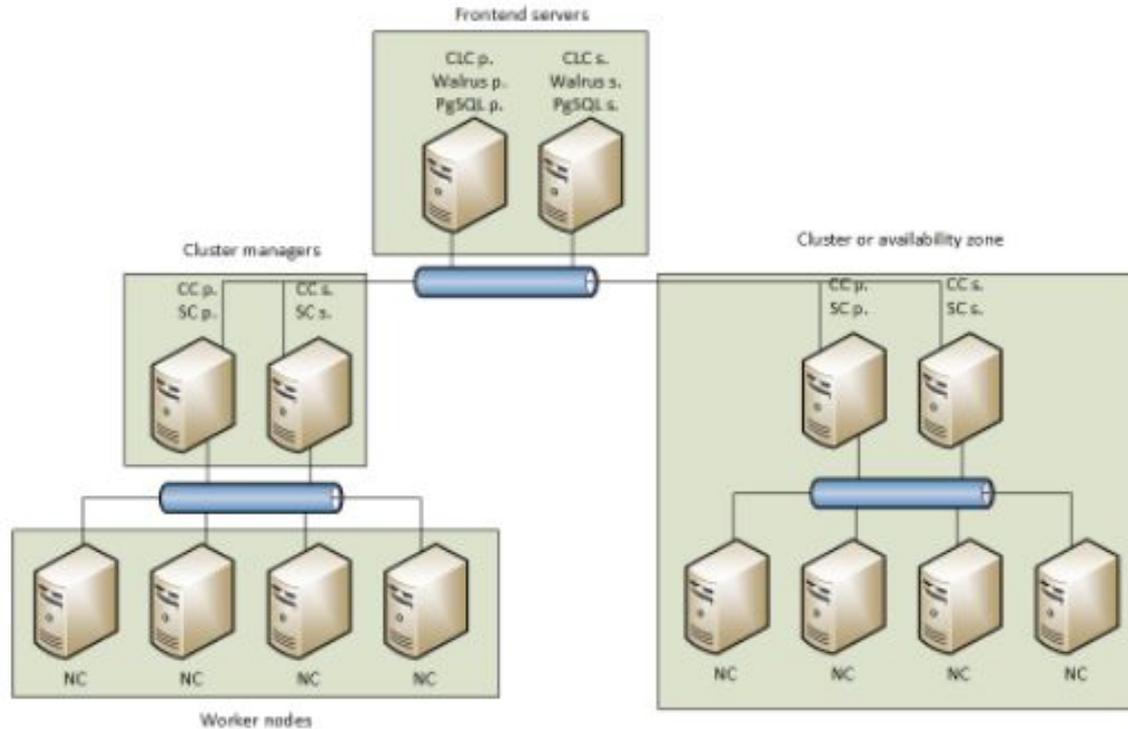


(e) Cold-standby SPARE



(f) Warm-standby SPARE

# Arquitetura Eucalyptus



# Modelos

- Modelo top-level

Número de blocos em paralelo como parâmetro

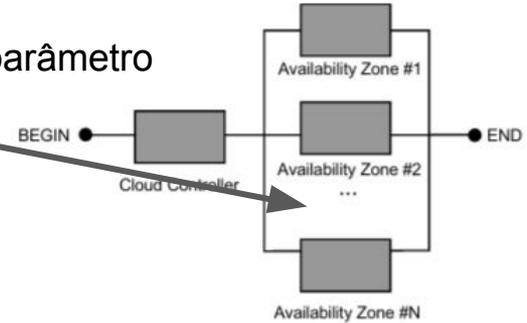


Fig. 12. Top level model of a private cloud infrastructure

```
RBD CloudModel{
  hierarchy cloud_controller(availability=solve(model = CloudController, metric = aval) );

  parallel avzones(
    times = av_zones,
    hierarchyBlock = (
      availability=solve_rm(model = ClusterModel, metric = aval)
    )
  );

  series s1 (cloud_controller, avzones);

  top s1;

  metric aval = availability;
}
```

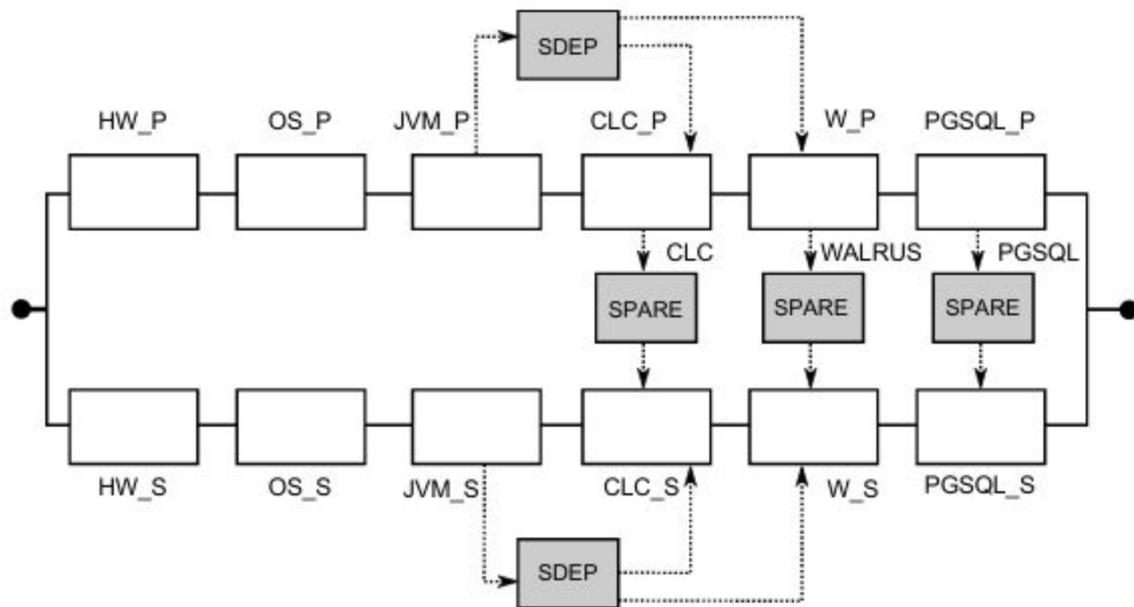


Fig. 13. Availability model for the Infrastructure Manager



Fig. 14. Availability zone model

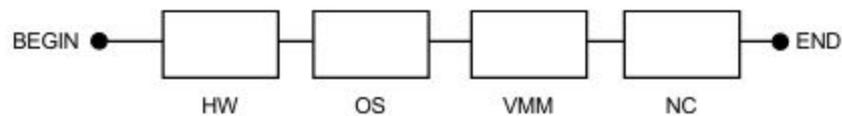


Fig. 16. Worker node model

# Avaliação de disponibilidade

- O cenário baseline exhibe 3 nove de disponibilidade, entretanto, há algumas possibilidades de melhoria:
  - Mais nós trabalhadores extras
  - Mais zonas de disponibilidade extras
  - Mais equipes de reparo
  - Aquisição de hardware mais confiável
  - Etc.

```
main{  
  
    aval = solve( model = CloudModel, metric = aval );  
    downtime = (1-aval) * 8760;  
    println( "Availability: " .. aval );  
    println( "Annual Downtime: " .. downtime );  
}
```

## Output

```
Availability: 0.9995442503348211  
Annual Downtime: 3.992367066967315
```

# Análise de sensibilidade

- Foram incorporados na linguagem funções para dois tipos de análise de sensibilidade
  - Design of Experiments
  - Diferenças percentuais
- Dado o grande número de parâmetros, aplicamos primeiro a segunda técnica para reduzir o número de parâmetros, e em seguida aplicamos a segunda. Os dois ranks obtidos permitem obter insights de como fazer upgrade na disponibilidade do sistema

```

percentageDifference(
  model_ = "CloudModel",
  metric_ = "aval",
  reevaluate = "true",
  samplingPoints = 5,
  parameters = (
    av_zones = [1,2,3,4,5],
    extra_workers = [0, 1, 2, 3, 4],
    repair_teams = [1,2,3,4,5],
    mttf_hw = [6000, 12000],
    mtrr_hw = [0.5, 4],
    mttf_os = [2000, 4000],
    mtrr_os = [0.25, 2],
    mttf_jvm = [2000, 4000],
    mtrr_jvm = [0.25, 2],
    mttf_service = [500, 2000],
    mtrr_service = [0.25, 3],
    mttf_vmm = [2000, 4000],
    mtrr_vmm = [0.25, 3],
    act_service = [0.001, 0.1],
    mttf_san = [1000000, 10000000],
    mtrr_san = [0.25, 3]
  ),
  output = (
    type = "swing",
    yLabel = "Steady-state availability",
    baselineValue = aval
  )
);

```

```

designOfExperiments(
  model_ = "CloudModel",
  metric_ = "aval",
  reevaluate = "true",
  parameters = (
    av_zones = [1,5],
    extra_workers = [0, 4],
    repair_teams = [1, 5],
    mtrr_hw = [0.5, 4],
    mtrr_os = [0.25, 2],
    mttf_service = [500, 2000],
    mtrr_service = [0.25, 3],
    mtrr_vmm = [0.25, 3],
    act_service = [0.001, 0.1],
    mttf_vmm = [2000, 4000]
  )
);

```

# Análise de sensibilidade

TABLE III

SENSITIVITY RANKING FROM PERCENTAGE DIFFERENCE

Parameter	Effect
extra_workers	0.01877449575191028
mttr_service	0.002150094848141864
act_service	8.013075495133587E-4
mttf_service	6.73011041538398E-4
mttr_vmm	4.2100187016546486E-4
av_zones	4.152073959725983E-4
mttr_os	3.072028528647158E-4
repair_teams	1.960734569628519E-4
mttr_hw	1.7903657686937967E-4
mttf_vmm	1.0527934744496536E-4
mttf_hw	6.483898160160282E-5
mttf_os	3.843260727462969E-5
mttr_jvm	2.193959602689862E-5
mttf_jvm	9.543993829277324E-6
mttf_san	8.999990999478921E-7
mttr_san	2.9815550166788506E-7

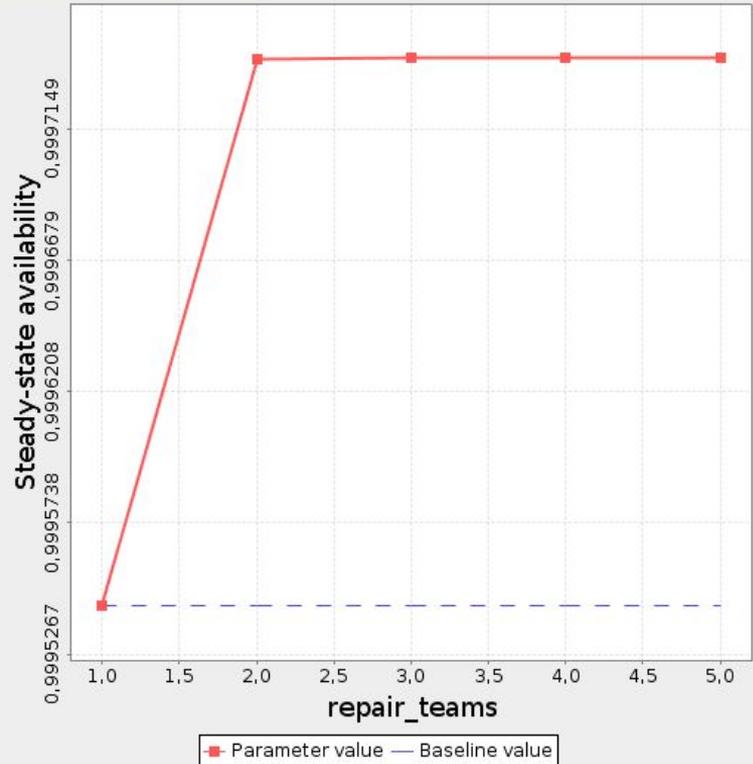
TABLE IV

SENSITIVITY RANKING FROM DOE

Parameter	Effect
av_zones	0.01635722120988814
extra_workers	0.016167285249836016
mttr_service	-0.008554586827862032
mttf_service	0.006234579411618091
mttr_vmm	-0.002528422448136114
mttr_os	-0.001503708217069466
mttf_vmm	9.9367098126657E-4
mttr_hw	-9.90086604369722E-4
act_service	-5.863656931597294E-4
repair_teams	1.812525480564542E-4

# Parameters

repair\_teams mtr\_hw mtf\_os mtf\_hw extra\_workers av\_zones

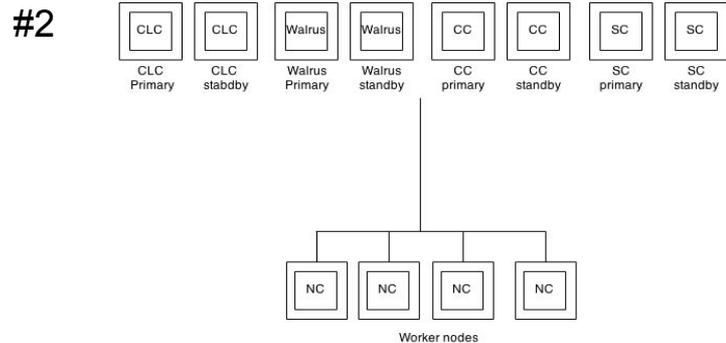
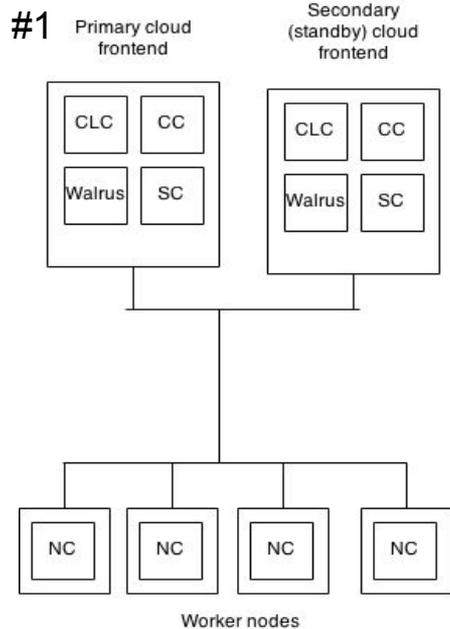


# Conclusões

- Aumentar o número de zonas de disponibilidade e nós trabalhadores extras provoca um salto imenso na disponibilidade. Porém, ter mais do que 2 nós trabalhadores extras, ou mais que uma zona de reparo extra, não influencia tanto assim na disponibilidade
- A técnica DOE explora um maior número de combinações de parâmetros, porém, utilizando apenas ela, não capturaríamos o insight anterior
- Não há motivo para não usar a técnica das diferenças percentuais, tendo em vista que é de rápida execução, pode ser usada para reduzir o espaço de parâmetros da técnica DOE, e permite visualizar nuances que não são visíveis pela técnica DOE

# Trabalhos futuros

- Otimização da plataforma Eucalyptus



**#3**

