Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas
Programa de Pós-graduação
em Computação Aplicada

CHAVE: Consolidation with High-Availability on Virtualized Environments

Daniel Scheidemantel Camargo

Orientador: Maurício Aronne Pillon, Dr.
Coorientador: Charles Christian Miers, Dr.

7 de dezembro de 2017

Agenda

- Motivação
- Trabalhos correlatos
- Solução proposta
- Plano de testes
- Resultados preliminares

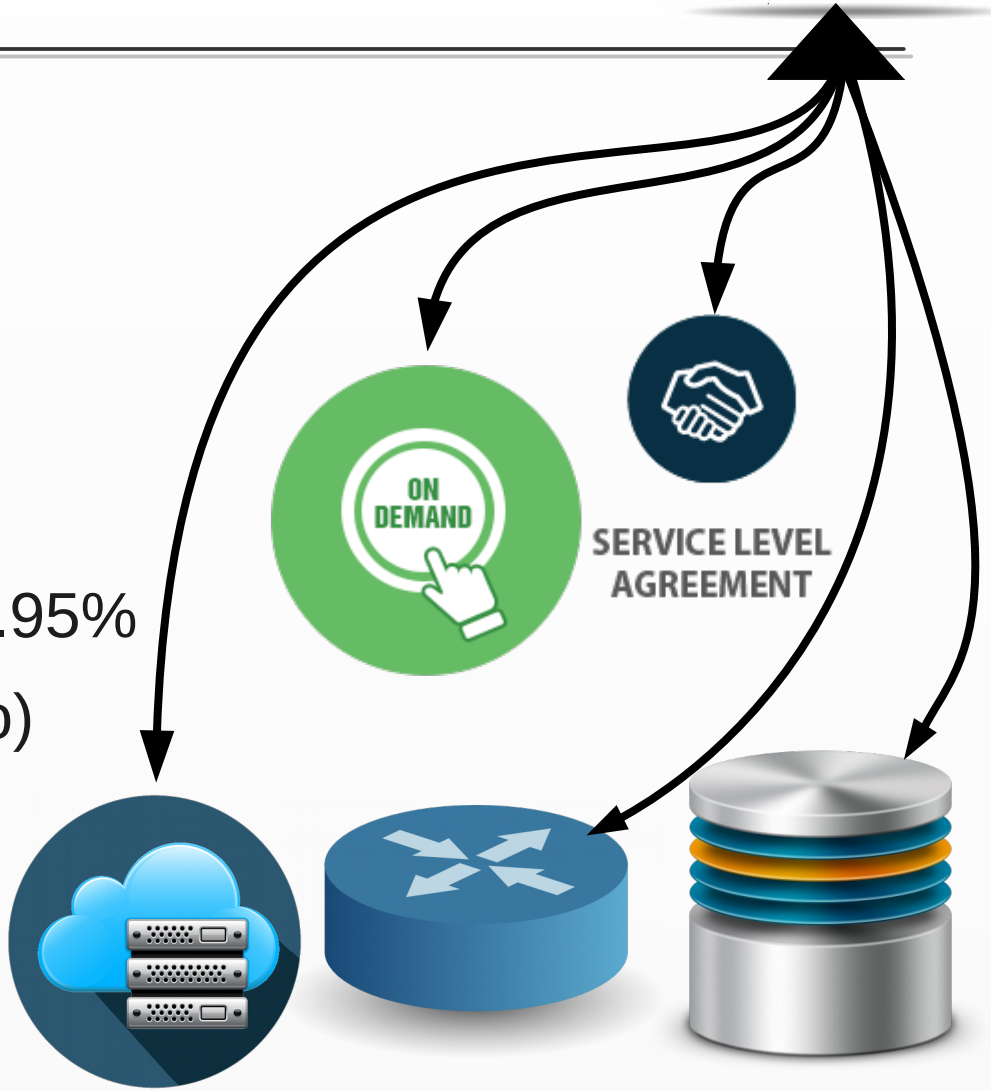
Objetivo

- Mecanismo de alta disponibilidade (HA):
  - Replicação de máquinas virtuais (MVs).
    - Impacto em energia e recursos
- Consolidação de MVs:
  - Reduzir o impacto da HA.
  - Restrição de justaposição das réplicas.

Computação em nuvem

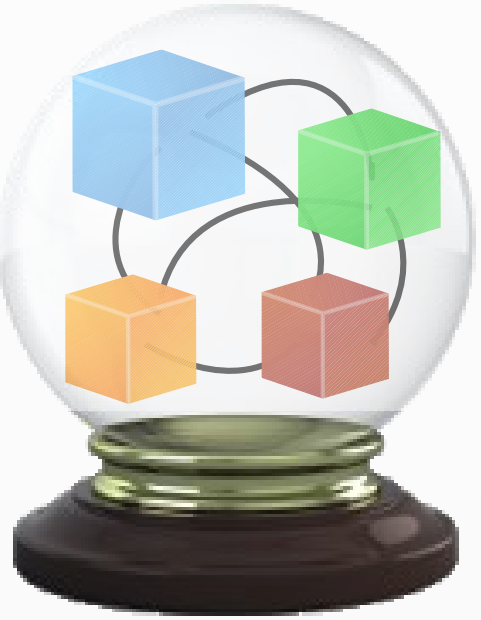
- Computação sob demanda.
- Virtualização
- SLA: **Disponibilidade**.
  - Nuvens públicas IaaS
    - Taxas: de 99.9% a 99.95%

      (131 a 66 minutos/ano)



Serviços críticos

- Organizações: **continuidade de negócios.**
- Violações no SLA (Interrupções)
  - Disponibilidade abaixo da contratada.
  - Prejuízos financeiro e reputação.
    - Ambos: cliente e provedor
- Se a taxa do SLA não atende aos requisitos da organização:
  - Serviços em alta disponibilidade (HA)
  - Acima de 5 noves (5.2 minutos/ano)



Disponibilidade

Inerente: sistemas computacionais distribuídos.
- Desconsidera-se logística e manutenção

- Razão entre tempo disponível (Ta) e tempo total (Tto)

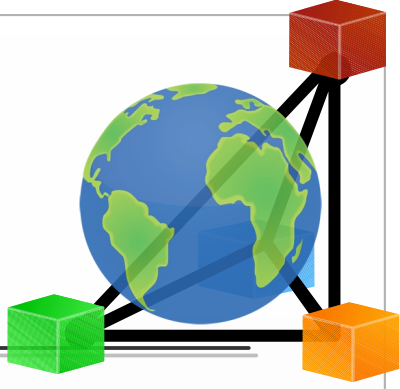
$$\mathbb{A} = \frac{T_a}{T_{to}} = \frac{T_a}{T_a + \sum_{i=0}^{n} T_{mc\ i}}$$

Nº de falhas

Tempo de correção

- Variáveis obtidas por **análise histórica.**
- Indicadores **probabilísticos**: MTBF, MTTF e MTTR.

Arquitetura Multi-AZs



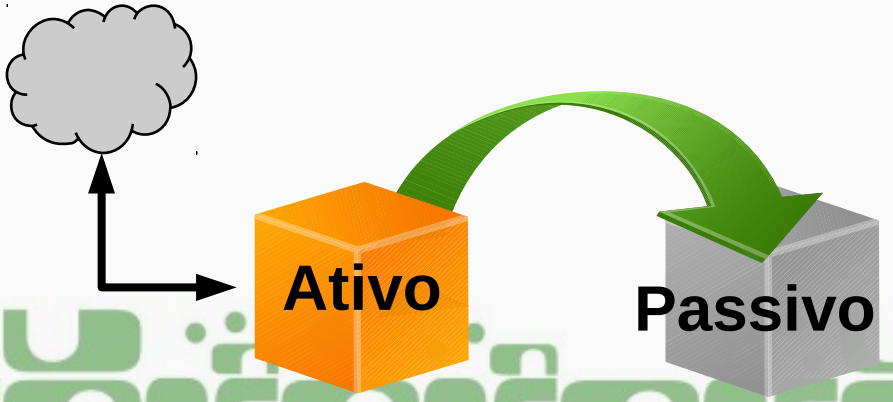
- São infraestruturas de data centers, agrupadas em regiões:
  - Globalmente distribuídas,
  - Independentes entre si (falhas independentes),
  - Links de baixa latência.
- Nuvens públicas e privadas.
- Multi-AZ amplamente usada para alta disponibilidade.

Alta Disponibilidade

- Serviços nativos para nuvem:
  - Balanceadores de carga autoescaláveis:
    - Sem estados em memória: ***stateless.***
  - Serviços autogerenciados:
    - SGBDs específicos.
- Demais aplicações ***stateful*** sem acesso a HA:
  - Serviços legados, aplicações científicas.

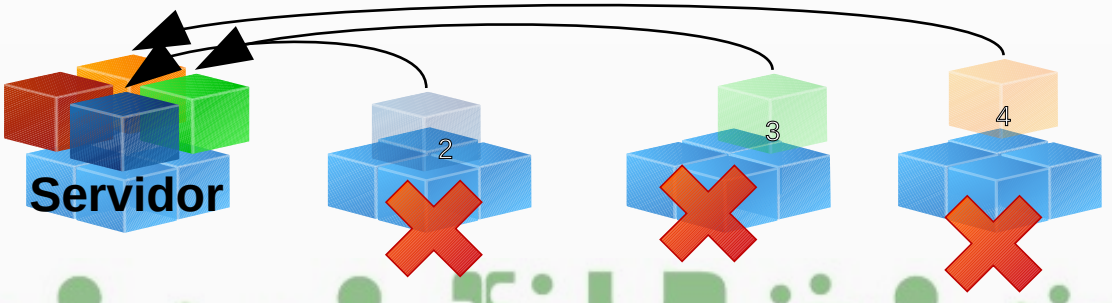
Alta Disponibilidade
Replicação

- Análise: *Tempo de correção* X *Sincronização* X *Recursos*.
  - Backup por *snapshots* (maior tempo de correção)
  - Replicação de estados de memória:
    - Ativo/ativo e ativo/passivo**.**
  - Maior **impacto no consumo de energia**



Consolidação de MVs

- Estratégia para reduzir o consumo de energia em DCs.
- Alocar todas as MVs no menor número possível de servidores, desabilitando os servidores subutilizados.
- Dilema: Consolidação *vs*. Violações do SLA.
- **Restrição de justaposição:** As réplicas de uma MV crítica não devem estar no mesmo ponto de falhas (SPOF)



Trabalhos correlatos

Trabalhos correlatos

- Relacionados doze trabalhos, organizados em três linhas de pesquisa:
  - Apenas consolidação, apenas HA, e ambos.
- Critérios:
  - **Contribuições:** Conceitos e práticas que agregam para a presente proposta.
  - **Problemas:** Conceitos incompatíveis com a presente proposta.

Trabalhos correlatos

- Linha de pesquisa:
  - Consolidação de MVs

Autores	Contribuições	Problemas
Nathuji e Schan 2007	Políticas locais e globais	Viola preceitos de isolamento e segurança
Corradi 2012	Mensura impactos negativos da consolidação	Sem uma estratégia definida
Beloglazov 2013	Utiliza o FFD para consolidar	Não considera diversas restrições
Zhang 2014	Correlaciona recursos físicos com virtuais	Utiliza apenas o posicionamento inicial

Trabalhos correlatos

- Linha de pesquisa:
  - Alta disponibilidade

Autores	Contribuições	Problemas
Ranjan 2015	Replicação Multi-AZ em nuvens públicas	Survey discute diversas abordagens.
Masakari 2017	Evacuação de MVs e reinício forçado	Apenas serviços *stateles*
Cully 2008	Usa *checkpoints* de replica de 1:N, tolerância à falhas	Compatível apenas com hipervisores Xen
Dong 2013	Abordagem em lockstep, proporciona melhor uso de recursos	Invasivo, pois requer modificações no TCP

Trabalhos correlatos

- Linha de pesquisa:
  - Consolidação de MVs com alta disponibilidade

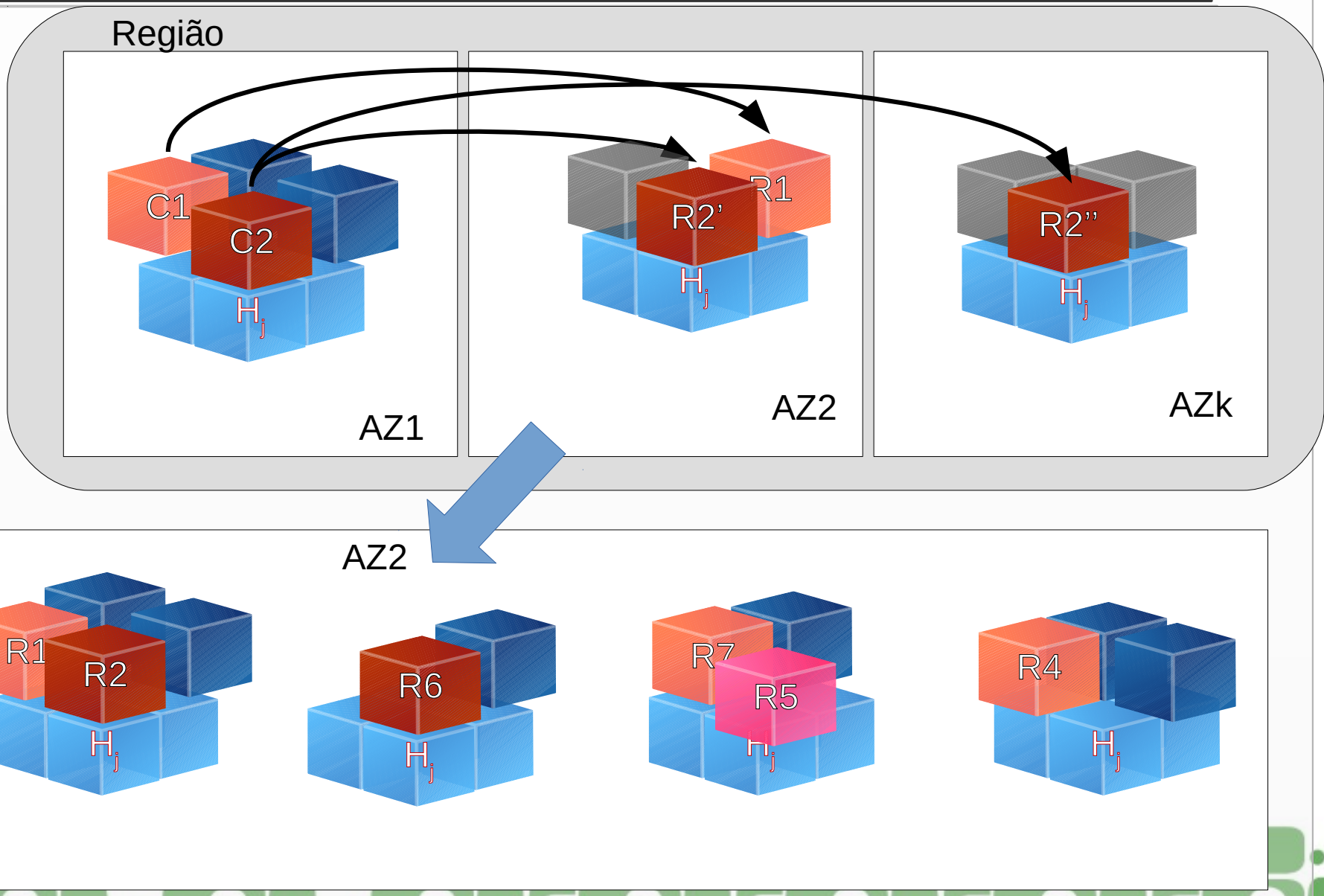
Autores	Contribuições	Problemas
Bin 2011	Adota a restrição de justaposição	*Backup* por *snapshots,* (tempo para restauração)
Simonin 2013	Arquitetura Big-Tree, verificação de falhas	Considera apenas +1 replica (sem HA OnDemand)
Li 2016	Replicação de 1:N, conceitos de probabilidade de falhas	*Backup* por *snapshots*, viola preceitos de isolamento e segurança

Trabalhos correlatos

- Diversas abordagens em consolidação ou HA:
  - Indicando que estas são potenciais áreas de pesquisa.
- Poucos trabalhos com as duas linhas de pesquisa.
- Levantamento das contribuições e problemas.
- Até o momento, não foram encontrados trabalhos que reúnam todas as características na mesma proposta.

CHAVE: Consolidation with High-Availability on Virtualized Environments

Objetivos gerais

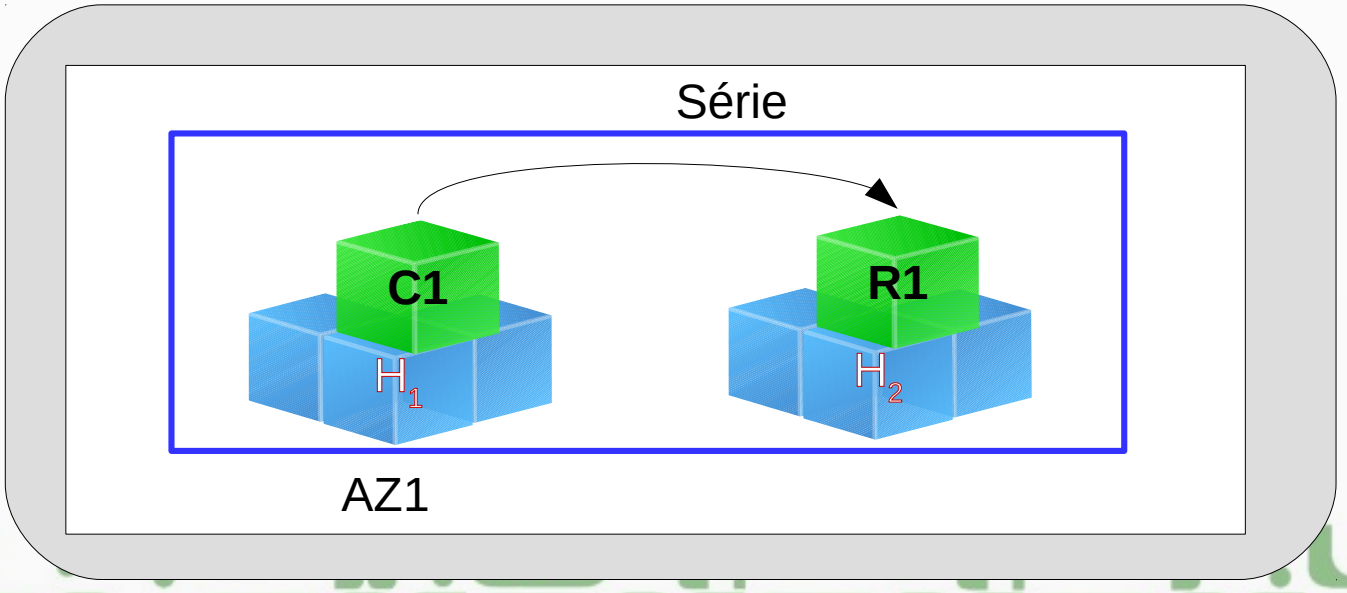


Arquitetura Multi-AZ

- As réplicas são obrigatoriamente instanciadas em AZs diferentes.
  - Minimiza o SPoF.
  - Habilita uma consolidação de MVs livre de restrições
    - Justaposição
    - Resultados mais próximos do ótimo

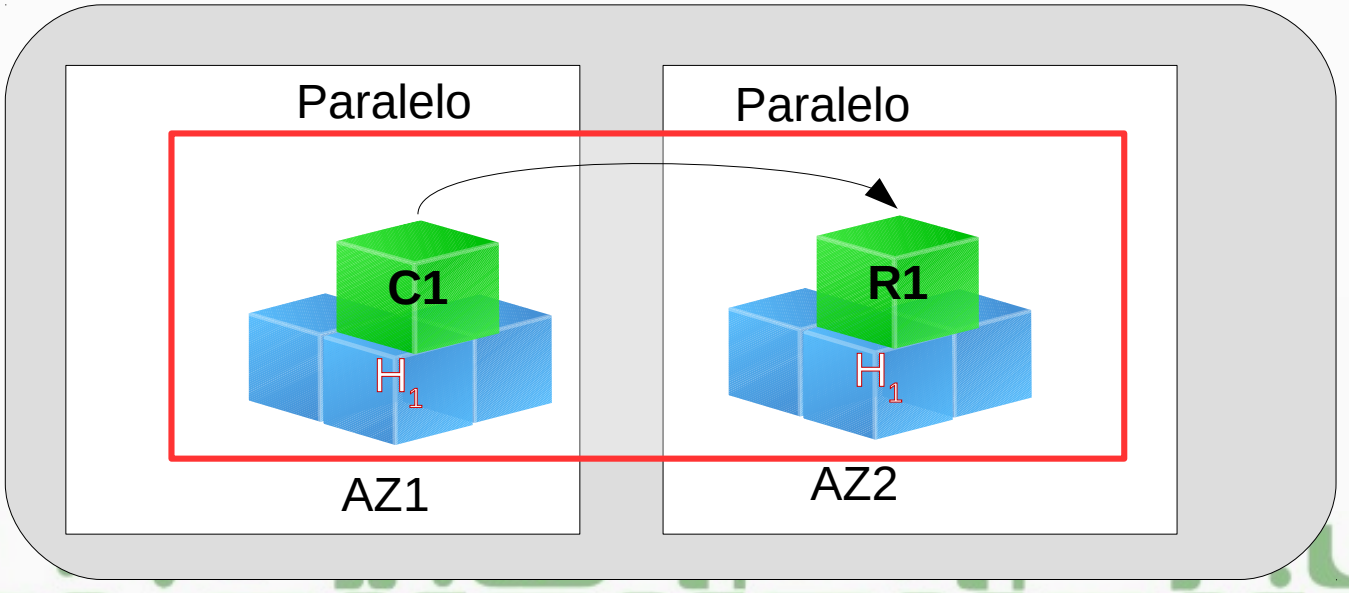
Alta Disponibilidade
Diagrama de blocos de confiabilidade

- Posicionamento das réplicas: componentes em série
  - Produto da disponibilidade da $AZ_{1,}$ servidor.
  - $A_1 * A_1 = A_{final}$
  - Resultado **menor** que a taxa de ambos.



Alta Disponibilidade
Diagrama de blocos de confiabilidade

- Posicionamento das réplicas: componentes em paralelo.
  - **1 -** Produto da **in**disponibilidade $(1-A_1)$, $(1-A_2)$,... $(1-A_n)$
  - $1 - (1-A_1) * (1-A_2) = A_{final}$
  - Resultado **maior** que a taxa dos componentes.



Alta Disponibilidade

- Probabilidade de falhas de eventos independentes.
- Para *k* componentes, a probabilidade de todas as AZs falharem simultaneamente:

$$\mathbb{HA} = 1 - (1 - \mathbb{A})^k$$

- '*k*' é o número de paralelos (1 MV crítica + réplicas)
- Entradas: Taxa de disponibilidade *A* e o valor de '*k*'

HA sob demanda

- Transparência para o desenvolvedor:
- Como saber o número de réplicas?
  - "*Forneça a taxa de HA requerida*". Ex.: 6 'noves'
  - A solução fornece o número de réplicas '*r*' necessárias para alcançar essa taxa.

$$\text{ceil}(r) \approx log_{1-\mathbb{A}}(1-\mathbb{HA}) - 1 \equiv \frac{log_{10}(1-\mathbb{HA})}{log_{10}(1-\mathbb{A})} - 1$$

- Base na disponibilidade $\mathbb{A}$ de cada AZ.

Definições Replicação

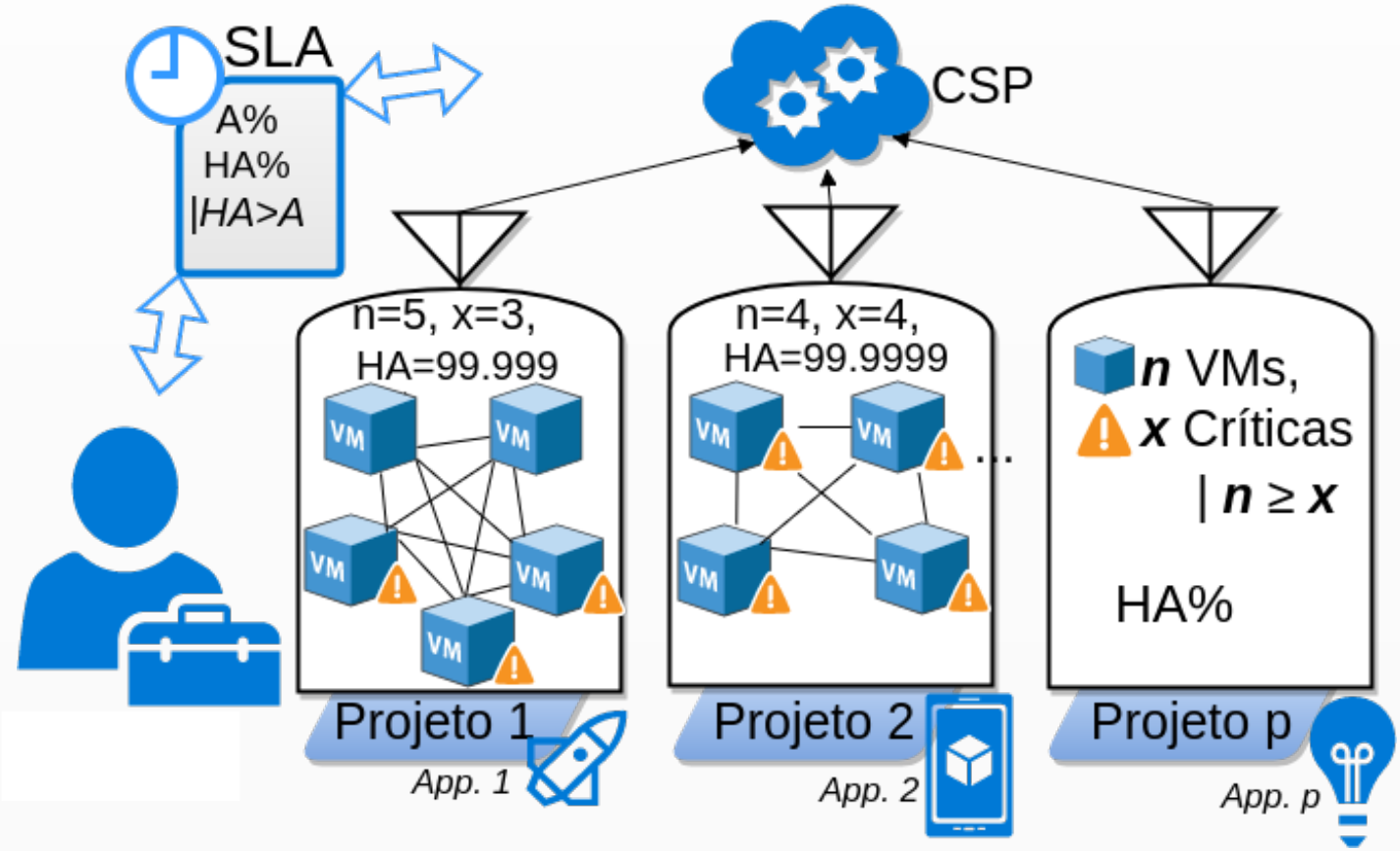
Perspectiva do desenvolvedor

-n MVs

-x Críticas: $\mathcal{V}_x^c$

-Taxa HA = XX%

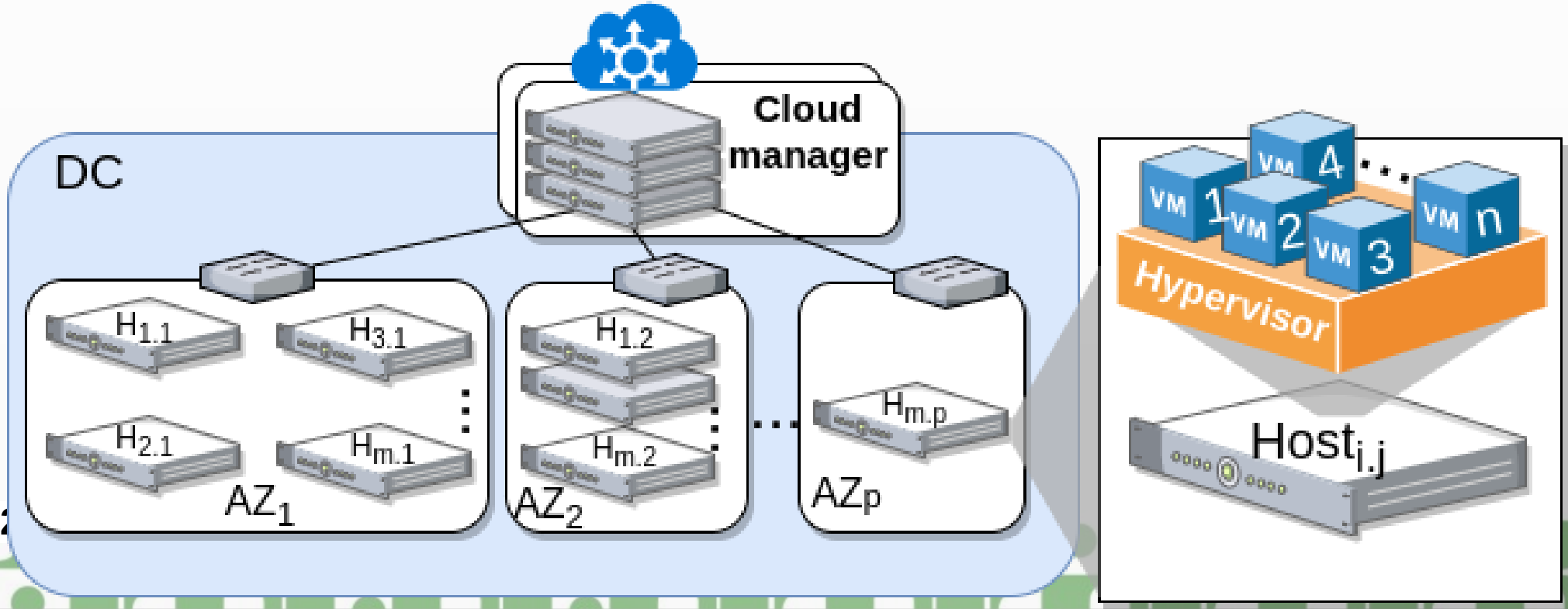
-Qual AZ ?



Definições Replicação

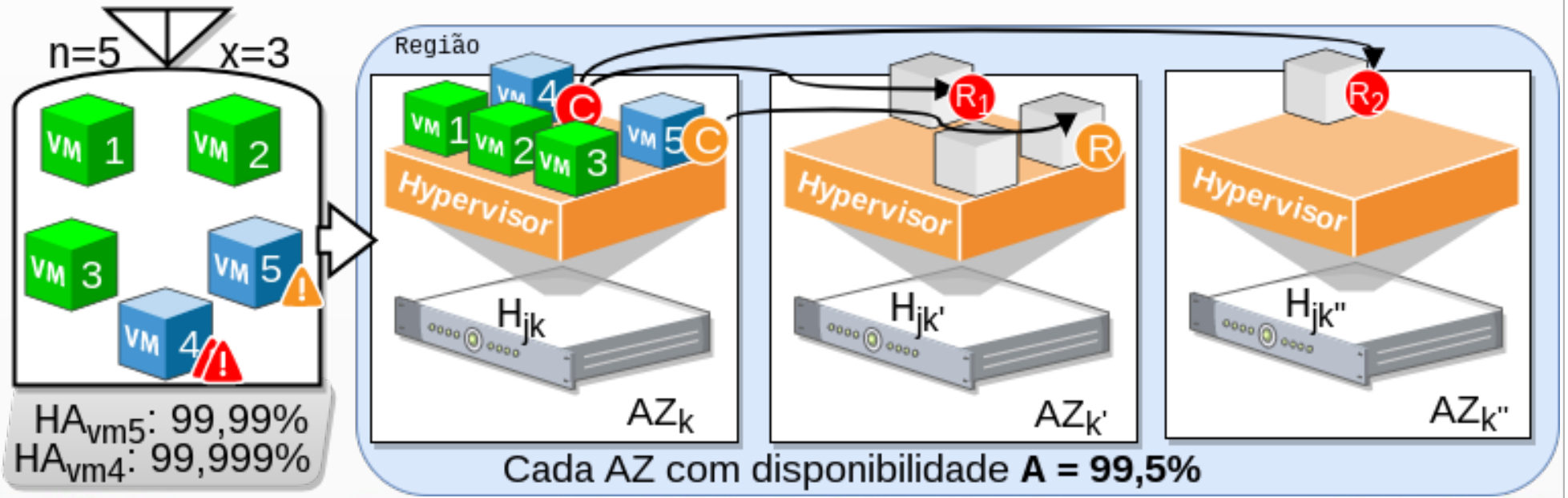
Perspectiva do provedor de nuvem

- ‘n’ MVs instanciadas: críticas e regulares. $\mathcal{V}_{nmp}$
- ‘m’ servidores em cada uma das AZs.
- Segurança e isolamento.

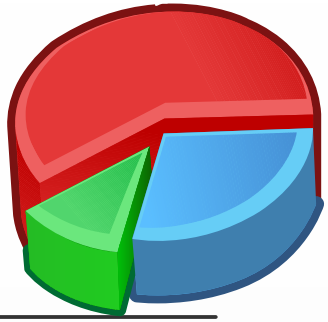


Definições Replicação

Conjunto de requisições de um mesmo desenvolvedor:
– VM5 demanda HA de **99,99%** = +1 réplica
– VM4 demanda HA de **99,999%** = +2 réplicas
– Disponibilidade regular **A = 99,5%**
– Duas MVs necessitam de três réplicas: Total 5 MVs

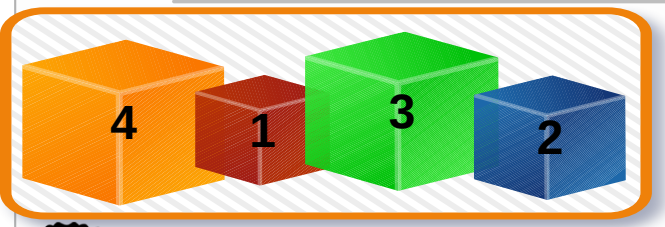


Consolidação de MVs

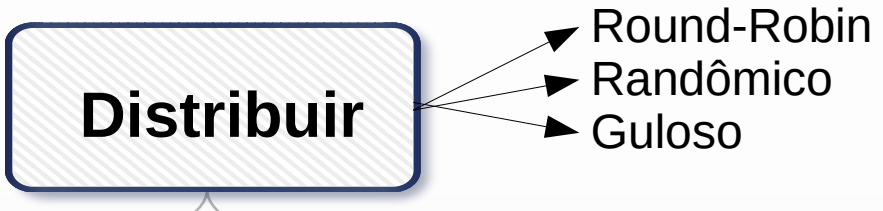


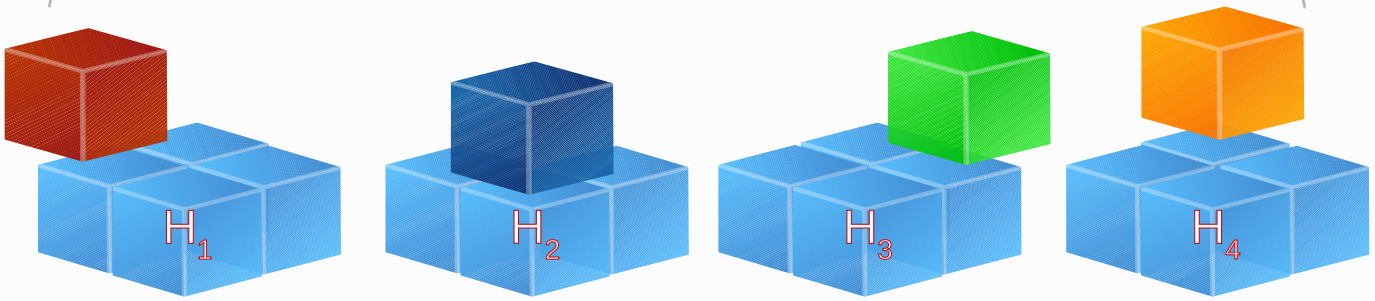
- Todas as MVs no menor número de servidores, desativando-os quando subutilizados.
- Problema do Bin-Packing (NP-Difícil).
  - Grande quantidade de requisições simultâneas inviabilizam a solução ótima por força bruta (tempo).
  - Heurísticas possibilitam obter uma solução aproximada em tempo hábil.
    - Relação entre tempo de resposta e otimalidade.
    - Algoritmo First-Fit Decreasing (FFD).
    - FFD(L) <= (11/9) * OPT(L) + n

Consolidação de MVs



Requisições FIFO



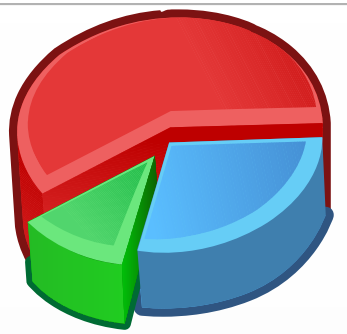


Consolidação ocorre em duas etapas independentes:

- Posicionamento inicial e Migração de MVs

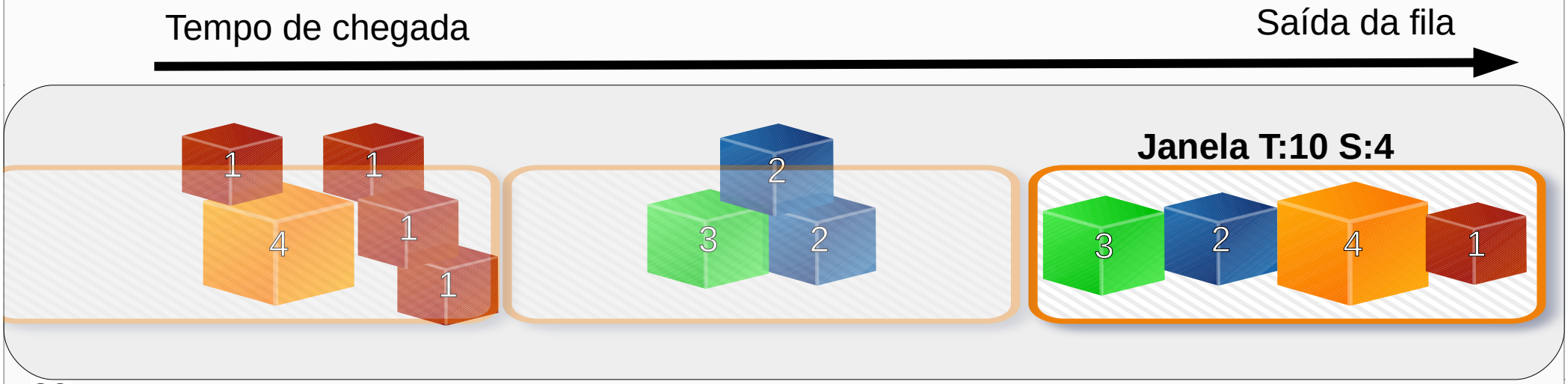
Consolidação de MVs

Posicionamento inicial



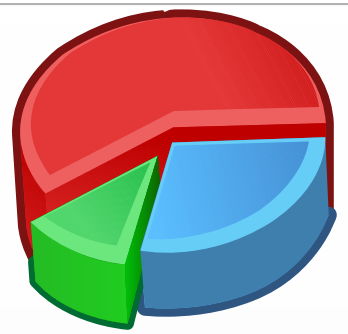
Utilizando o conceito de **janelas** com limitações de **tempo** e **tamanho:**

- Reduz a espera e complexidade.
- Abordagem *offline*



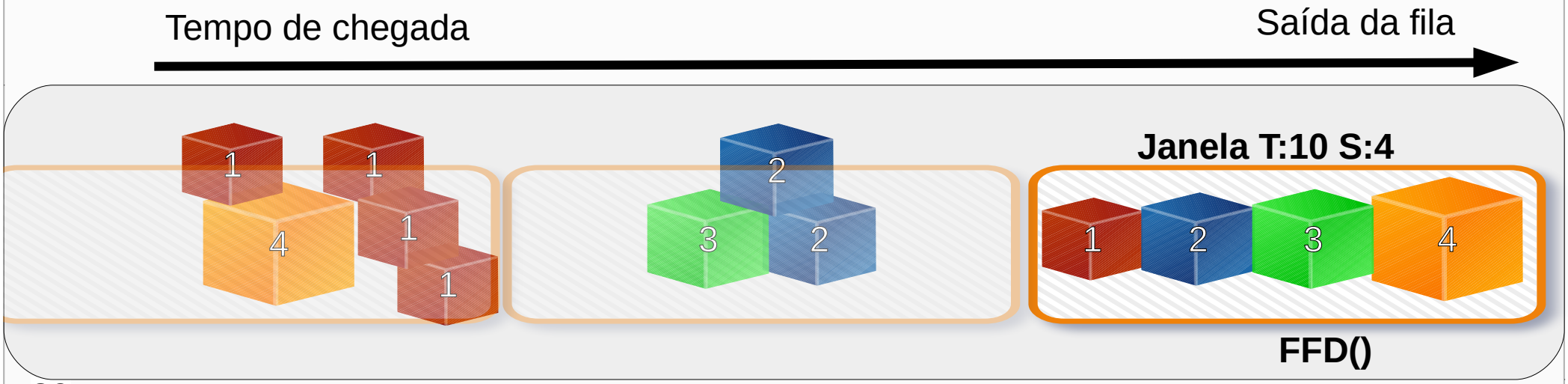
Consolidação de MVs

Posicionamento inicial



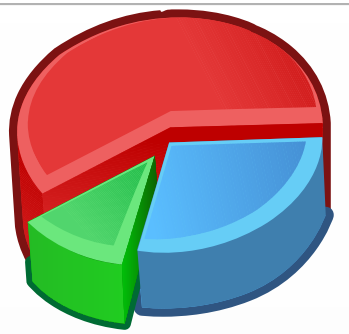
Utilizando o conceito de **janelas** com limitações de **tempo** e **tamanho:**

- Reduz a espera e complexidade.
- Abordagem *offline*

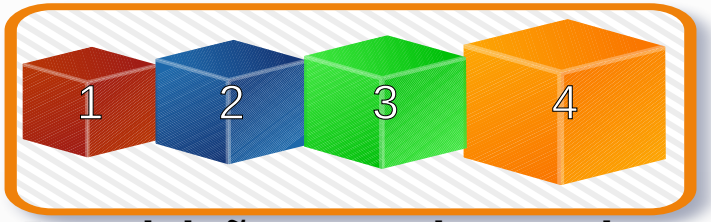


Consolidação de MVs

Posicionamento inicial



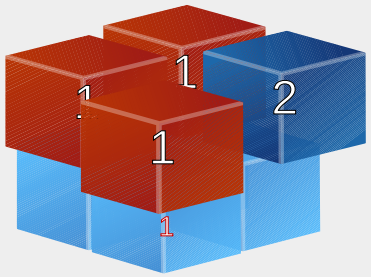
Reduzir complexidade na busca pelo melhor servidor:
- Ordenar servidores por recursos disponíveis (D).



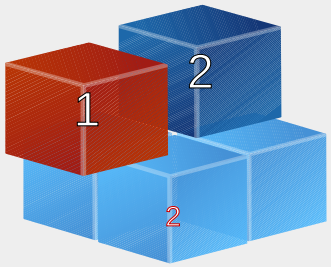
Requisições para instanciar

Ordenação dos servidores por recursos disponíveis **D**

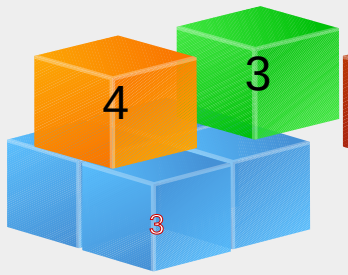
Servidores
Tamanho S=8



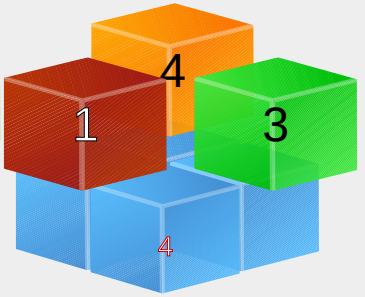
T=5,
D:3



T=3,
D:5



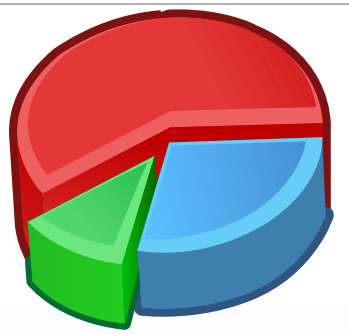
T=7,
D:1



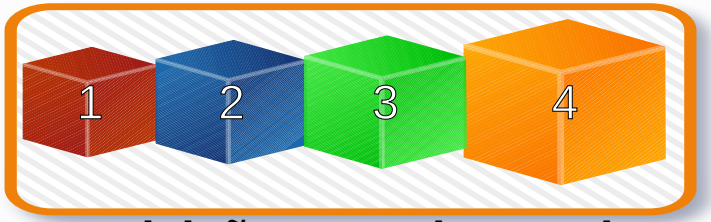
T=8,
D:0

Consolidação de MVs

Posicionamento inicial

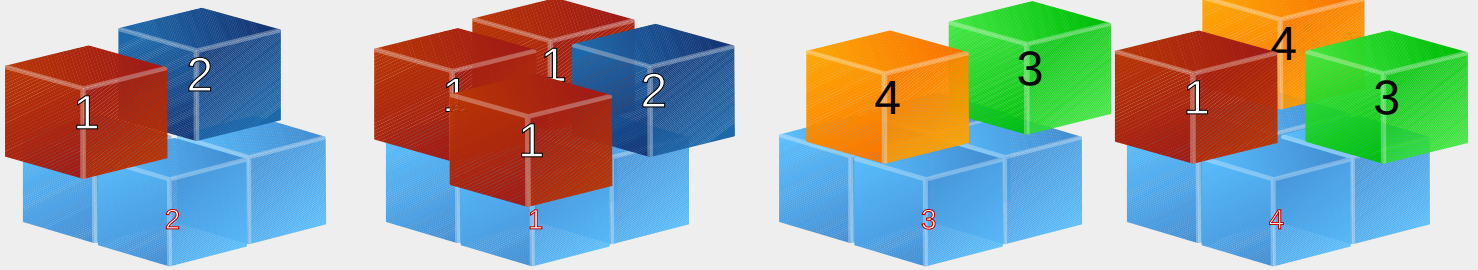


FFD Dynamic Decrasing (FF3D): Ordena **decrescente** os recursos disponíveis dos servidores



Requisições para instanciar

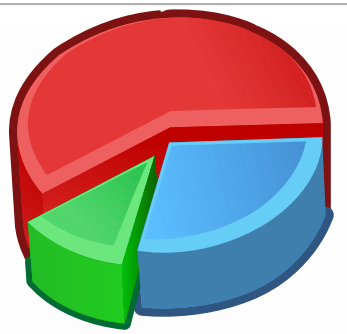
FF3D ( )
S=8



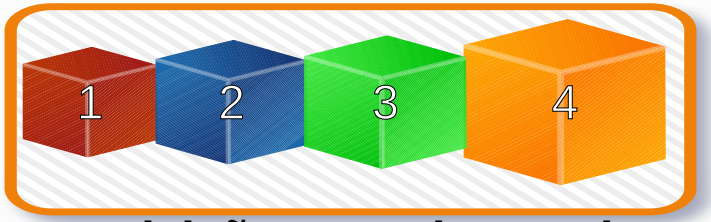
D:5 **D:3** **D:1** **D:0**

Consolidação de MVs

Posicionamento inicial

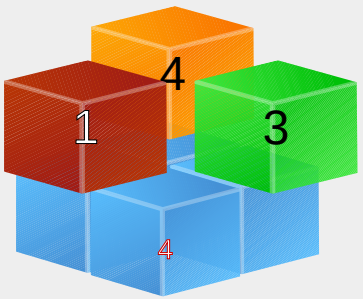


FFD Dynamic Decrasing Increasing (FF2DI): Ordena **crescente** os recursos disponíveis dos servidores

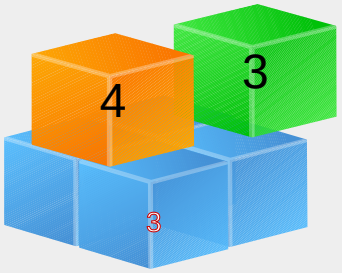


Requisições para instanciar

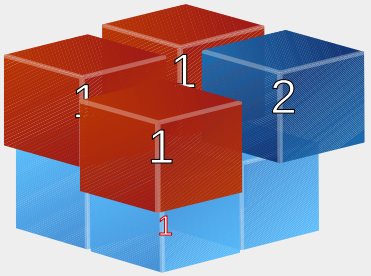
FF2DI( )
S=8



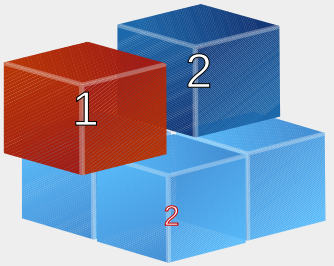
D:0



D:1



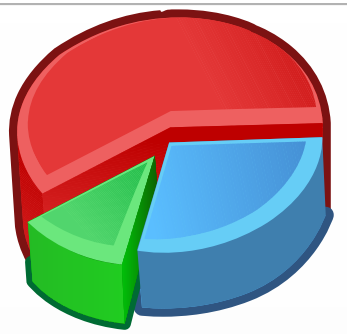
D:3



D:5

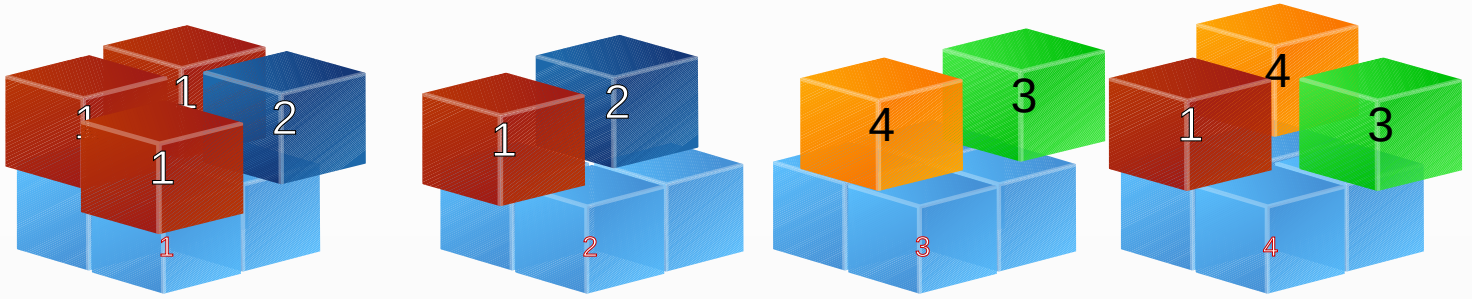
Consolidação de MVs

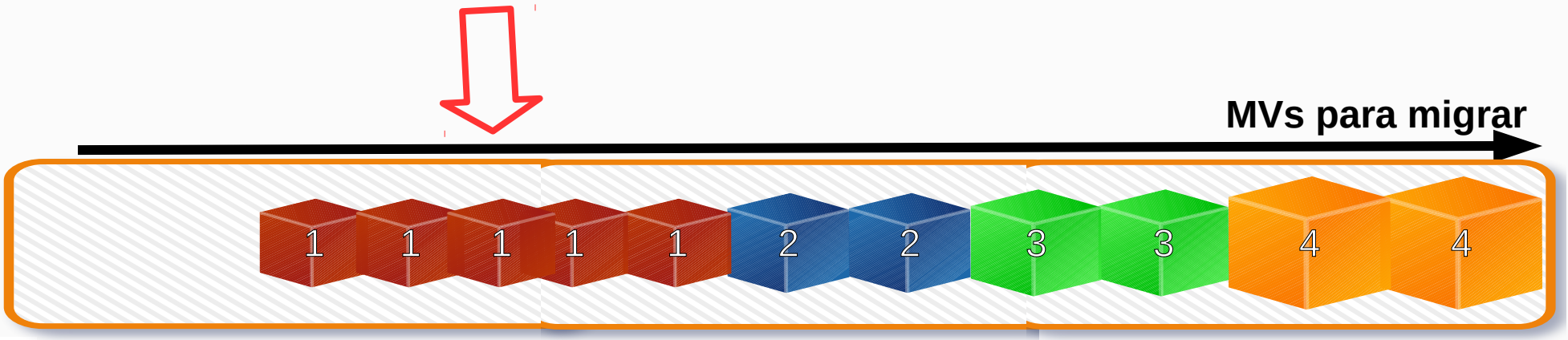
Migração de MVs



Muda as MVs após alocadas em um servidor (desativa-os).

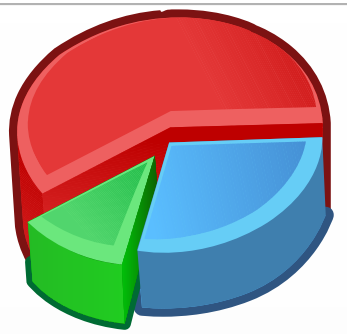
Simula a melhor configuração e migra o necessário.



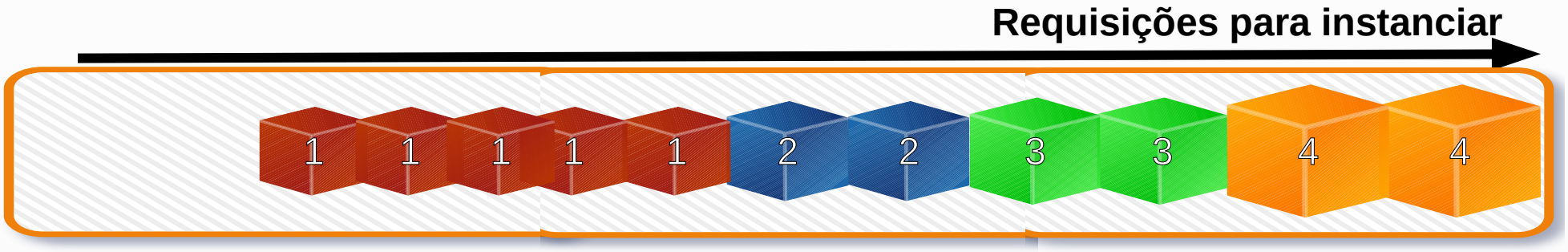


Consolidação de MVs

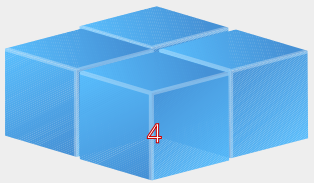
Migração de MVs



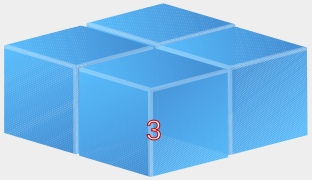
Cria-se uma lista de requisições fictícia



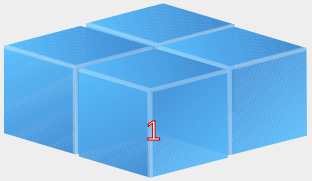
migra( )
S=8



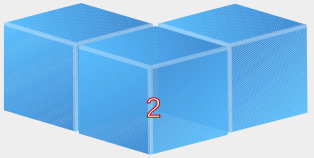
D:8



D:8



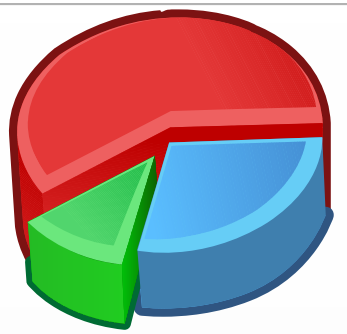
D:8



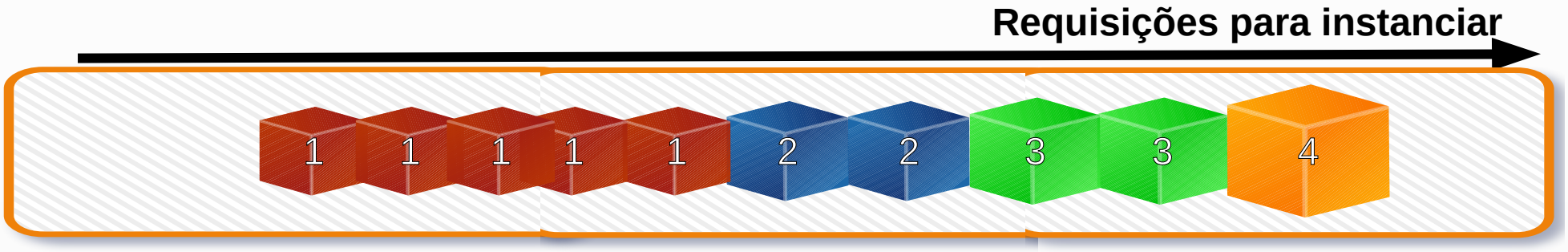
D:8

Consolidação de MVs

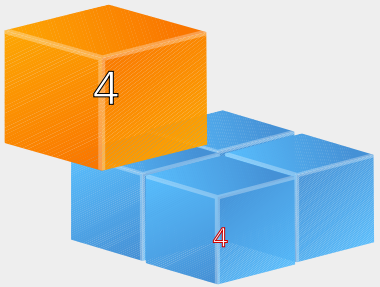
Migração de MVs



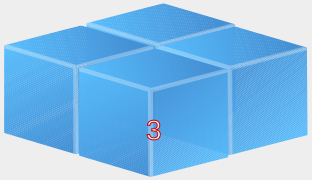
Cria-se uma lista de requisições fictícia



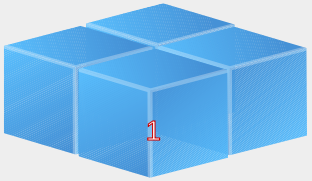
migra( )
S=8



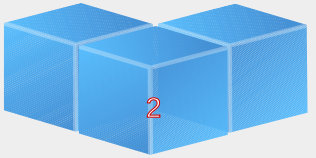
D:4



D:8



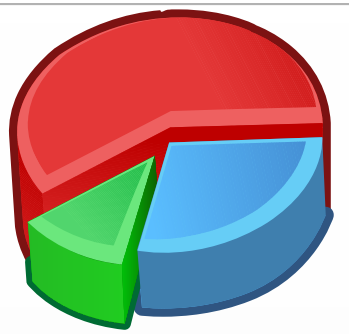
D:8



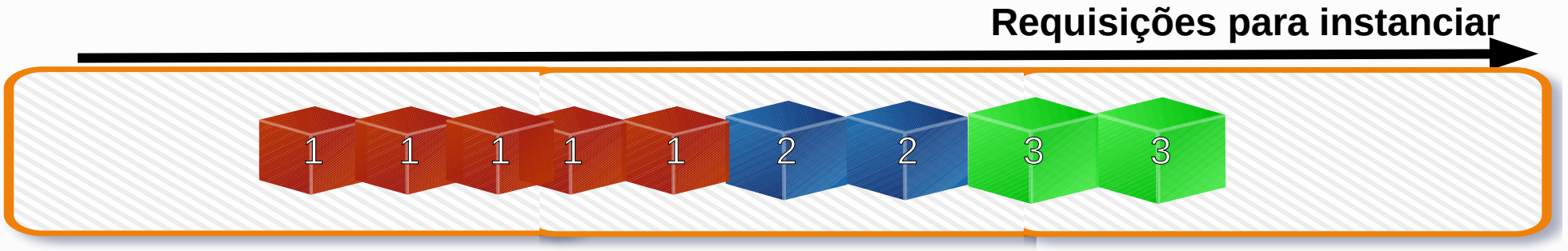
D:8

Consolidação de MVs

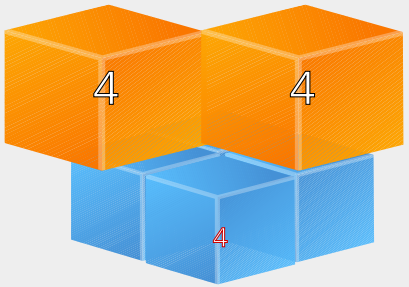
Migração de MVs



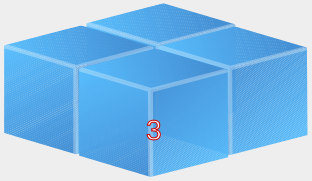
Cria-se uma lista de requisições fictícia



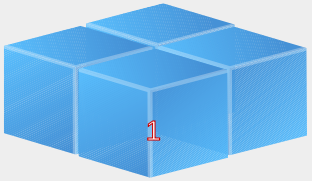
migra( )
S=8



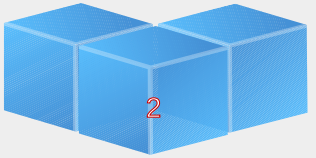
D:0



D:4



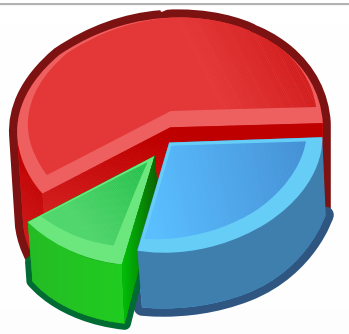
D:8



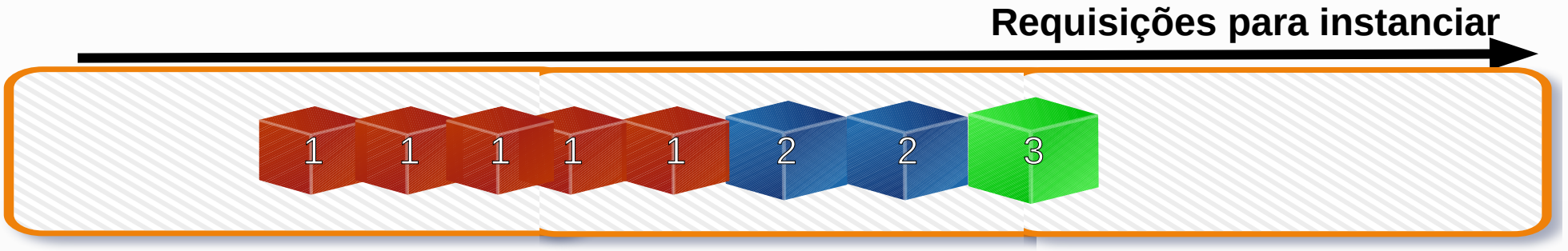
D:8

Consolidação de MVs

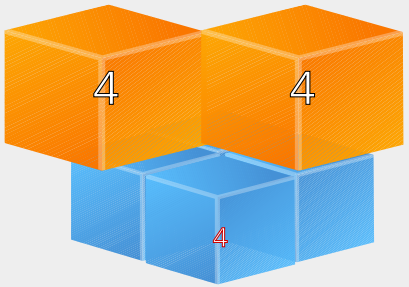
Migração de MVs



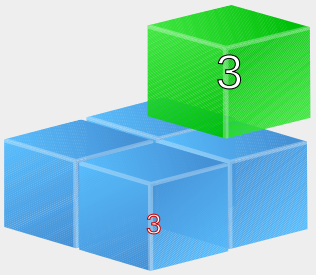
Cria-se uma lista de requisições fictícia



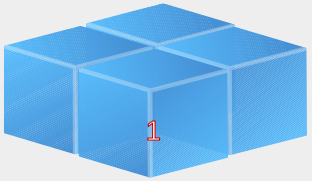
migra( )
S=8



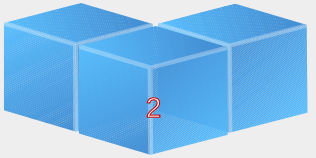
D:0



D:5



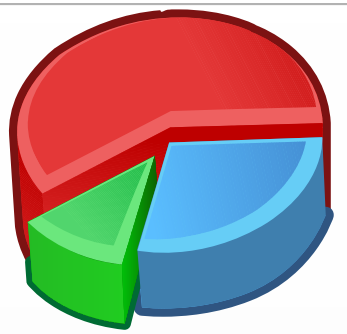
D:8



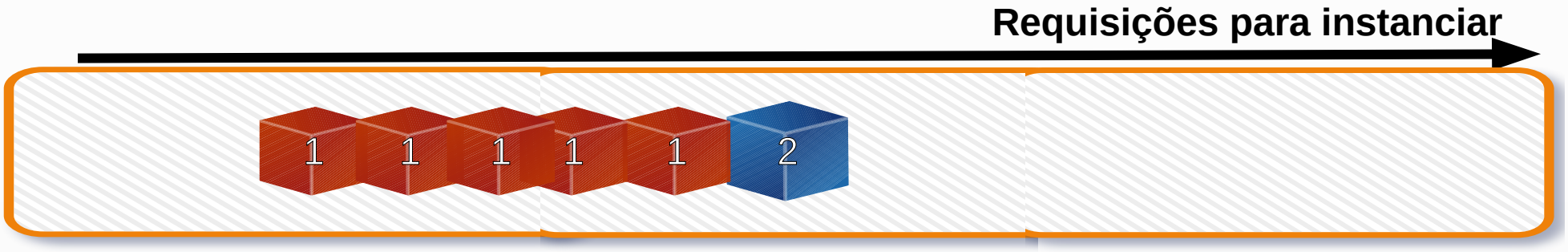
D:8

Consolidação de MVs

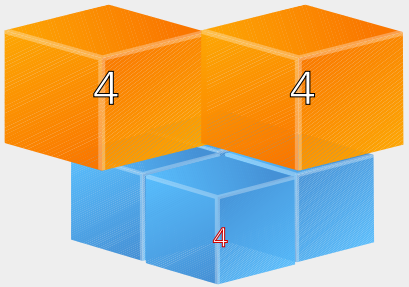
Migração de MVs



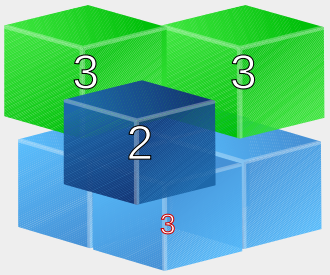
Cria-se uma lista de requisições fictícia



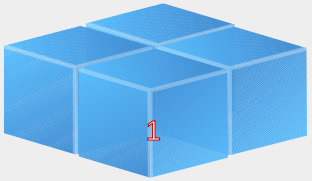
migra( )
S=8



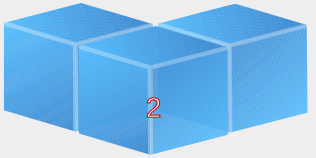
D:0



D:0



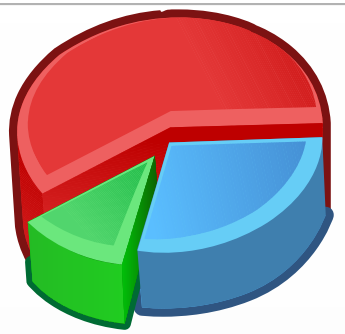
D:8



D:8

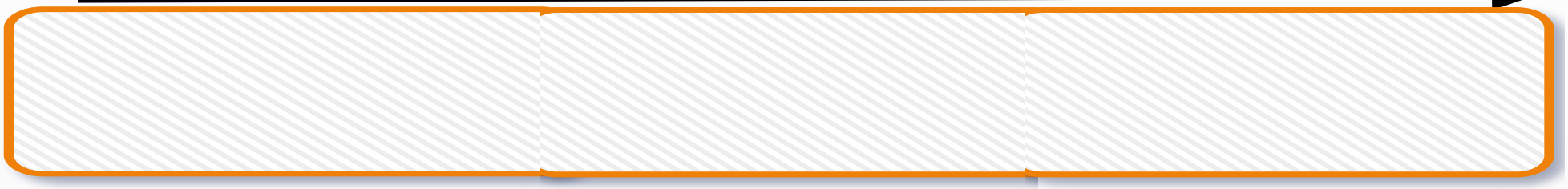
Consolidação de MVs

Migração de MVs

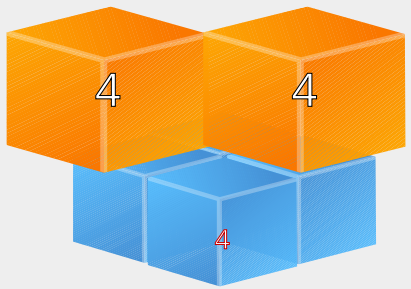


Cria-se uma lista de requisições fictícia

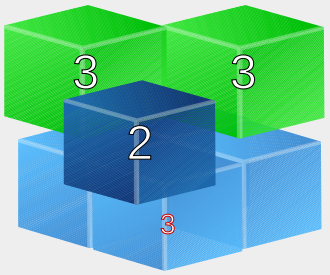
Requisições para instanciar



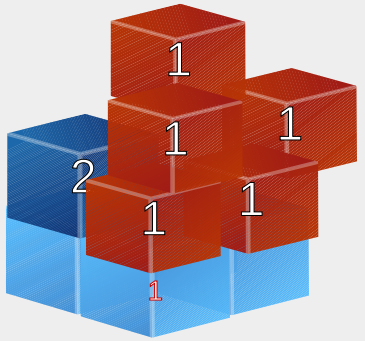
migra( )
S=8



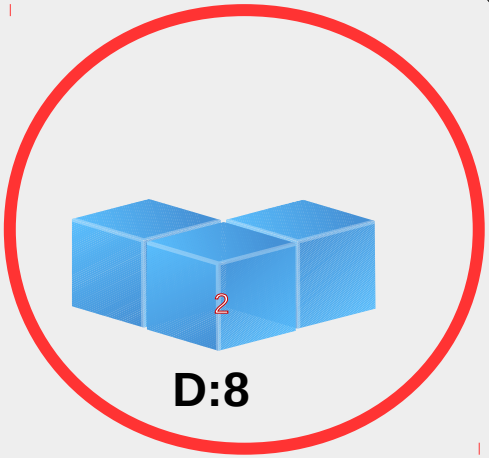
D:0



D:0



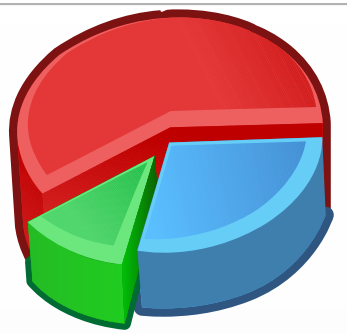
D:1



D:8

Consolidação de MVs

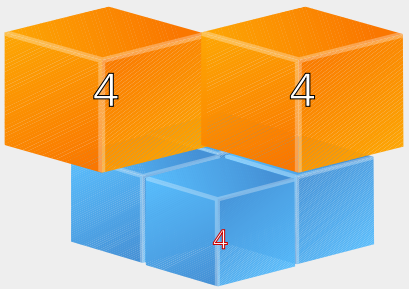
Migração de MVs



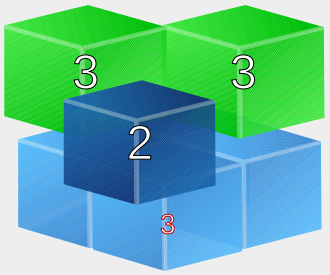
Cria-se uma lista de requisições fictícia

Requisições para instanciar

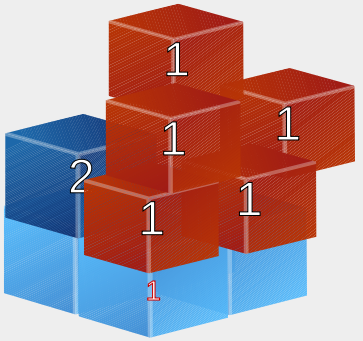
migra( )
S=8



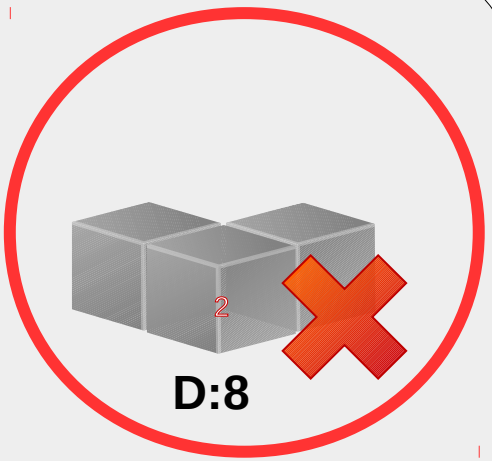
D:0



D:0



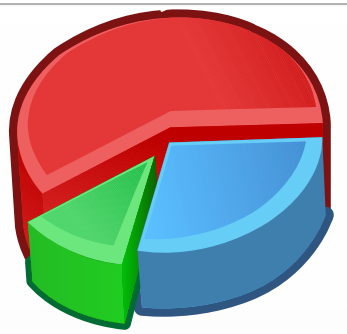
D:1



D:8

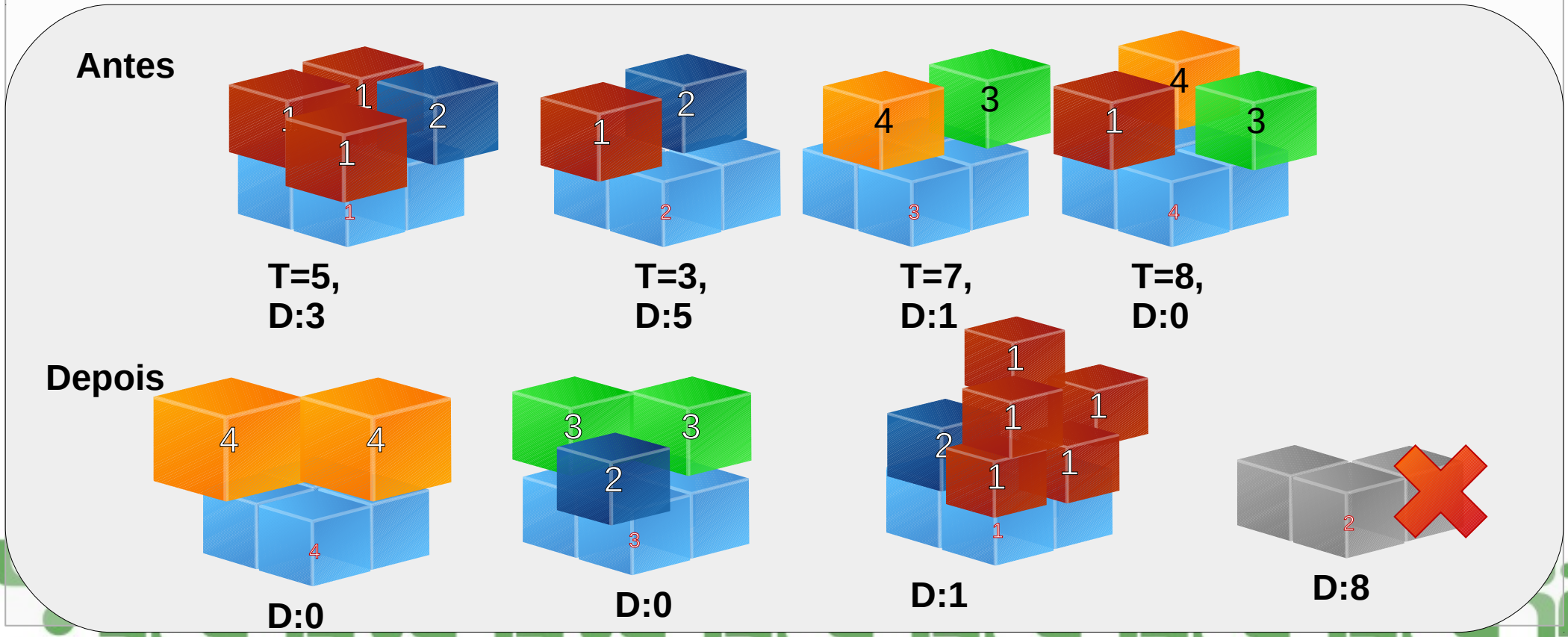
Consolidação de MVs

Migração de MVs



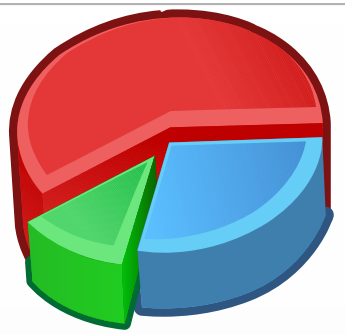
Número alto de migrações:

- Pior caso é o tamanho fila

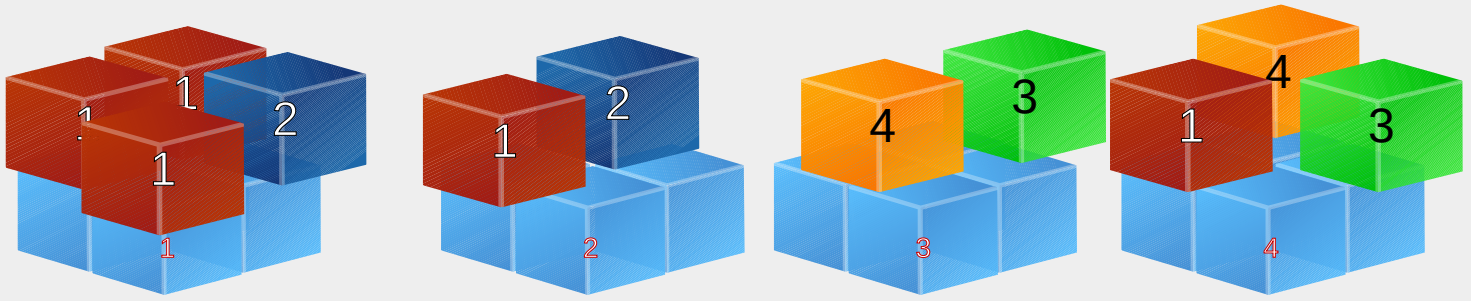


Consolidação de MVs

Migração de MVs



- *Minimum Migration Aware* (MMA): Visa minimizar o número de migrações



T=5, D:3

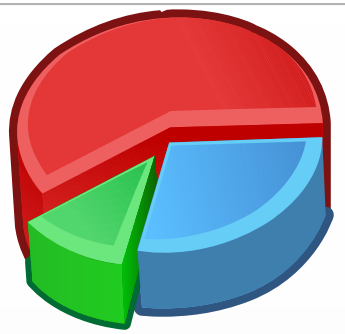
T=3, D:5

T=7, D:1

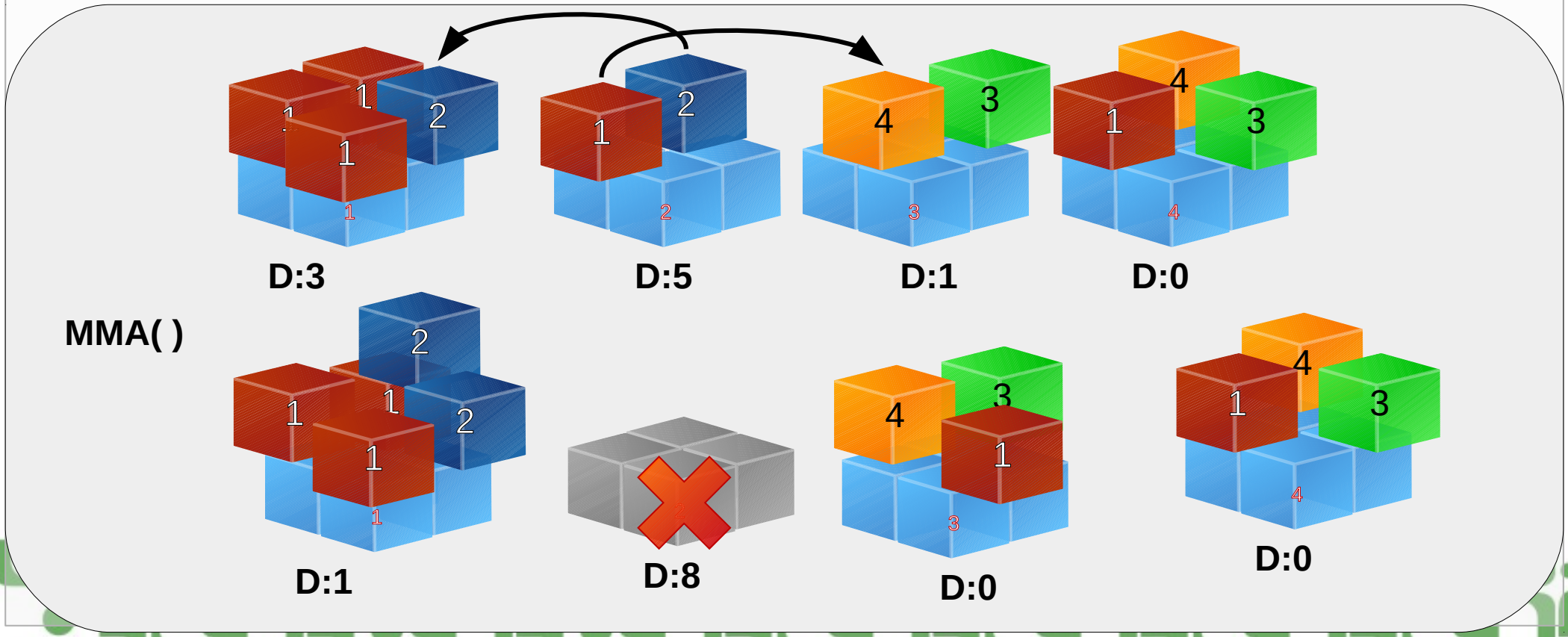
T=8, D:0

Consolidação de MVs

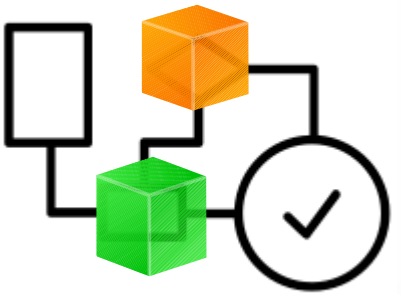
Migração de MVs



- *Minimum Migration Aware* (MMA): Visa minimizar o número de migrações

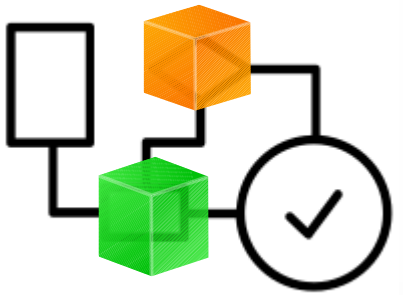


Conjunto de soluções



- Entre tantas possibilidades, qual a melhor abordagem?
  - Geral:
    - Migra antes de posicionar
    - Posiciona antes de migrar
  - Posicionamento:
    - FF3D ou FF2DI
  - Migração:
    - Total ou MMA
- Tamanho e tempo entre janelas, *overbooking,* etc

Conjunto de soluções

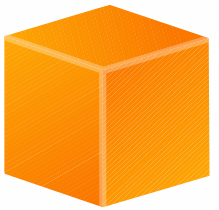


- Melhor combinação depende do conjunto de requisições!
  - Realiza numericamente todas as possibilidades:
    - Menor número de servidores ativos (energia).
    - No menor custo de tempo e de migrações.

- Determina-se a sequência de etapas a serem executadas pelo escalonador da nuvem.
- Solução agnóstica à plataforma de computação em nuvem.
  - Ações por meio de API.

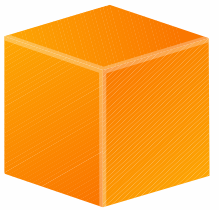
Estudo de caso

Estudo de caso



- Realizado no simulador EAVIRA (Ruck, 2017):
  - Linguagem Python
  - Alterações de infraestruturas virtuais para MVs.
- Executadas em uma instância na AWS, Ubuntu 16.04 com 4 vCPU e 30 GB RAM (memória otimizada).
- Resultados apresentados com base em *traces* do Eucalyptus.

Estudo de caso



- *Traces* reais de uma nuvem privada IaaS.
- Plataforma Eucalyptus (Wolski e Brevik, 2014).
  - 6 AZs, de 7 a 31 servidores de 8 a 32 núcleos cada
  - Mínimo de 1,8k até 18k operações:
    - < START / STOP >
    - Entre 33 e 279 dias de *trace.*
  - *Escalonamento: Greedy* e *Round-Robin*
  - Adicionada funcionalidades de overbooking (200%)

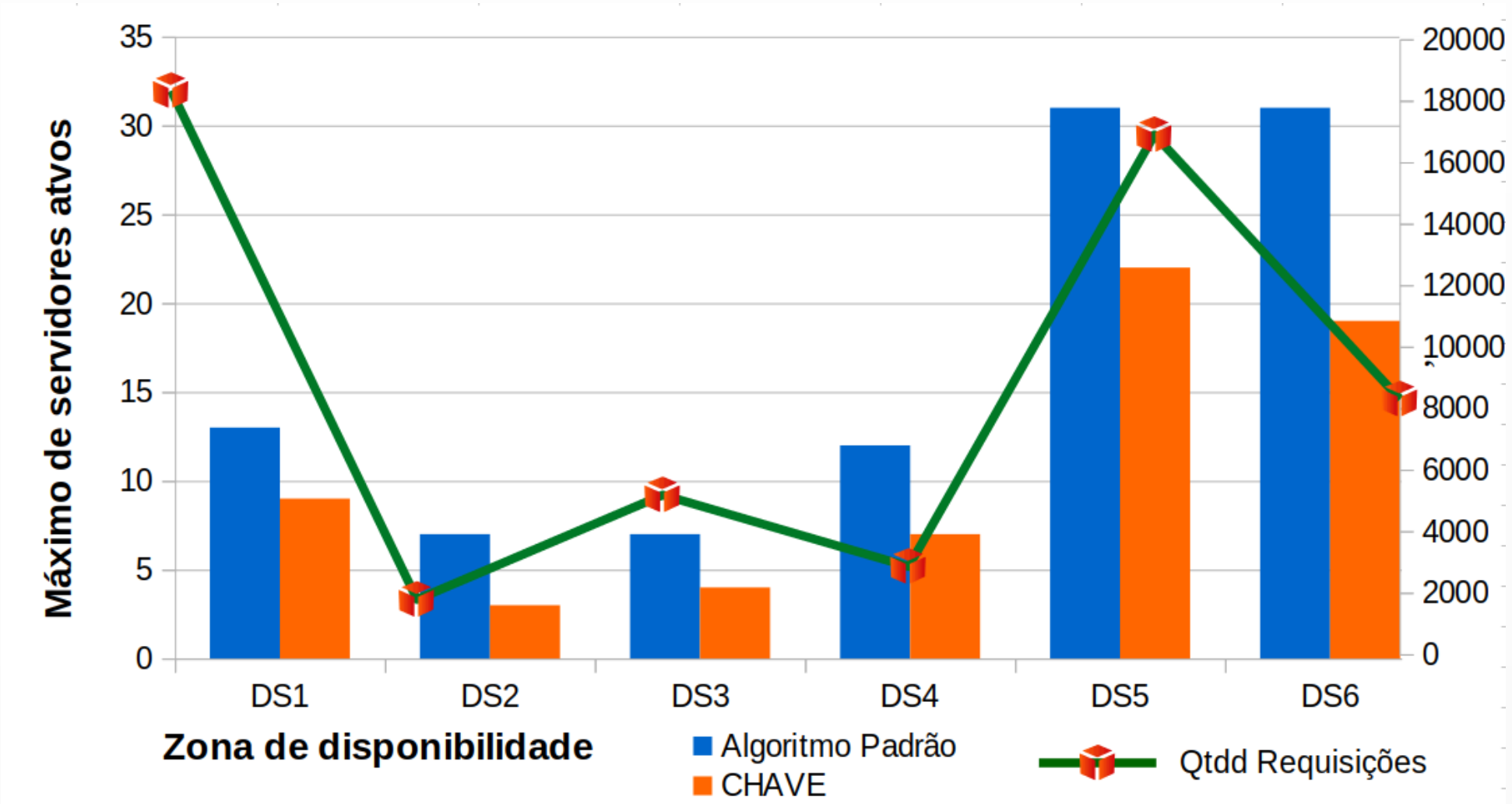
Plano de testes
Seleção das estratégias

Estratégia	Posiciona-migra	migra-posiciona	FF2DI	FF3D
E_0	*Sem consolidação*			
E_1	✓		✓	
E_2	✓		✓	
E_3	✓			✓
E_4	✓			✓
E_5		✓	✓	
E_6		✓	✓	
E_7		✓		✓
E_8		✓		✓

Plano de testes

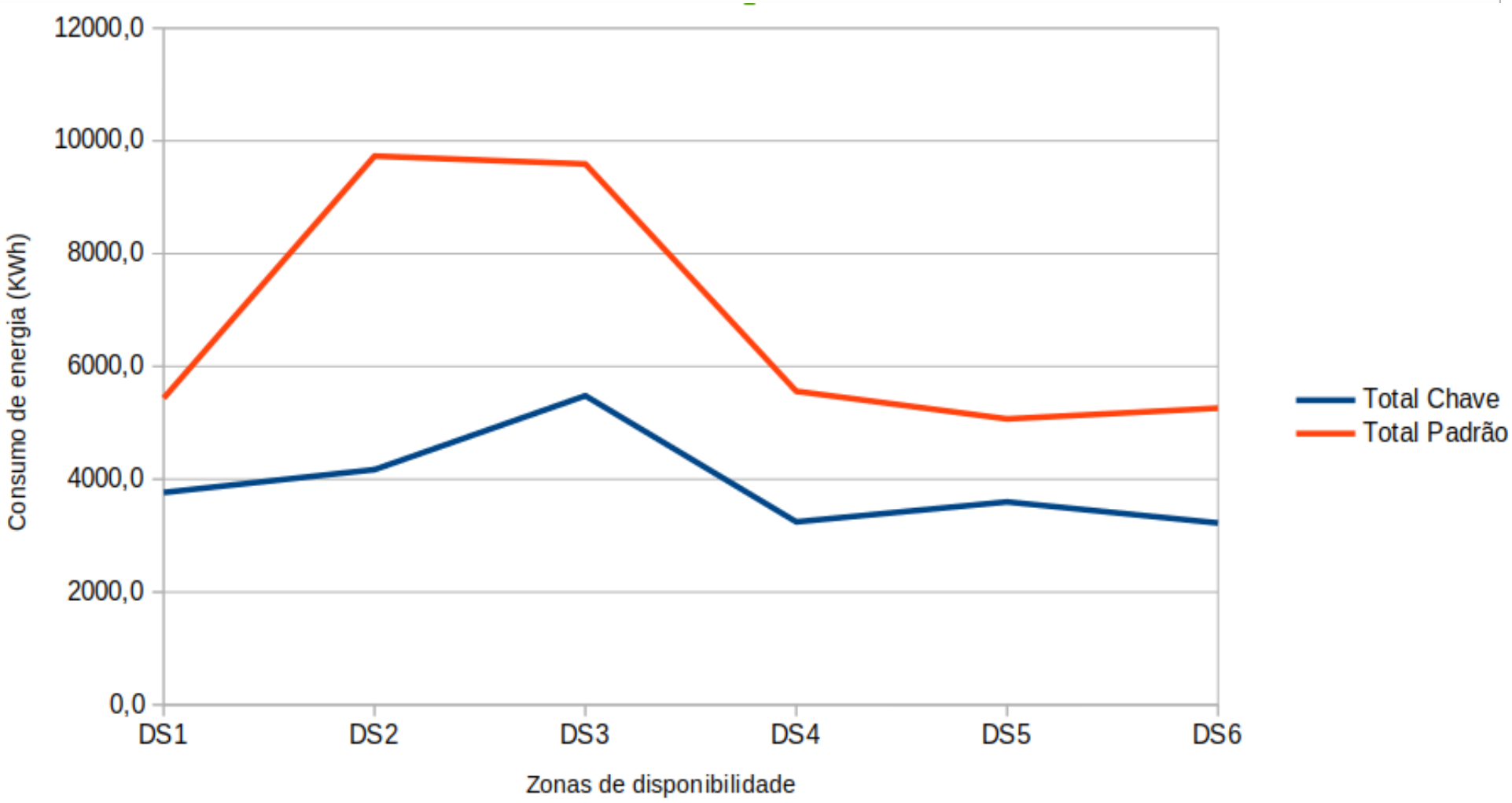
- Tamanho das janelas: 1% do conjunto de requisições.
- Tempo entre janelas: de 20 a 200 segundos.
- Para cada algoritmo: CHAVE e padrão do Eucalyptus:
  - O maior número de servidores ativados ao mesmo tempo? (Pior caso)
  - Consumo de energia total para o período de execução?
  - Relação final de eficiência?
  - Quantas réplicas é possível instanciar com o CHAVE, mantendo o mesmo consumo de energia dos algoritmos padrão do Eucalyptus?

Resultados:
Quantidade de servidores



38,7% desativados

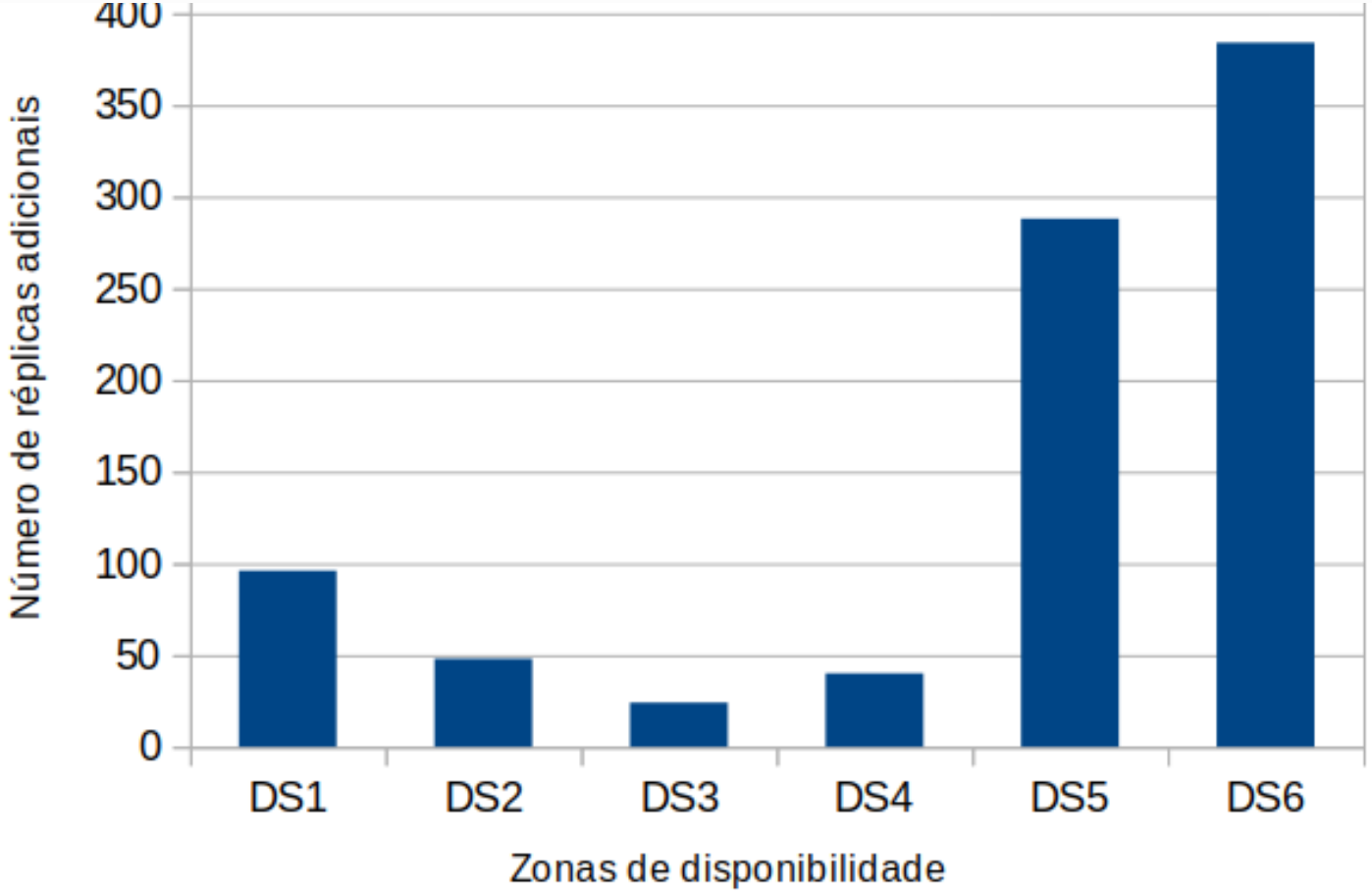
Resultados:
Consumo de energia



Eficiência final de 42,2%

Resultados: Réplicas adicionais

- Quantas réplicas adicionais poderíamos adicionar em cada AZ para manter o mesmo consumo de energia?



(#P - #C) * Cores

Min: 24
Max: 380

Total: 880 MVs

Replicas Adicionais

Discussão dos resultados

- Foi possível reduzir a quantidade de servidores em média de 40% no pior caso de consolidação
- Obtida uma eficiência energética de 42,2% no total, apenas com a consolidação.
- Ao realizar a replicação, para manter o mesmo consumo de energia poderiam ser replicadas até 880 MVs críticas no total, com uma réplica cada.
- As melhores configurações entre seleção de estratégias de algoritmos foram varias, pois depende das requisiões.

Considerações gerais

- A presente solução objetiva a fornecer alta disponibilidade sob demanda utilizando a consolidação de MVs para reduzir o impacto causado pelo consumo de energia.
- Trabalhos relacionados indicam oportunidades na área de pesquisa. Poucas soluções que unem HA e consolidação.
- Devido a complexidade de implementar a solução em um ambiente real, é utilizada a simulação, mas o objetivo final é aplicar o CHAVE em um estudo de caso em ambiente real OpenStack.

Próximos passos

- Realização de mais testes considerando todas as possibilidades levantadas.
- Gerar *traces* aleatórios, distribuições variadas.
  - Outros *traces* reais.
- Implementar a consolidação em ambiente real OpenStack.
- Adequar o simulador para ser utilizado em outras pesquisas
  - Documentação, didática, entrada e saída de dados, etc.

Publicação de Artigos

- Publicados:
  - Daniel Camargo, Maurício Pillon, Charles Miers Guilherme Koslowski "***MeHarCEn: Método de Harmonização do Consumo de Energia em Data Centers",*** RBCA 2017.
  - Durante a graduação: Consumo de energia em DCs.
- Em submissão:
  - Daniel Camargo, Maurício Pillon, Charles Miers Guilherme Koslowski, ___. "***CHAVE: Consolidation with High-Availability on Virtualized Environments***", SBRC 2018.

Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas
Programa de Pós-graduação
em Computação Aplicada

CHAVE: Consolidation with High-Availability on Virtualized Environments

Daniel Scheidemantel Camargo

daniel@colmeia.udesc.br