

# Selecionando Provedores de Computação em Nuvem usando Indicadores de Desempenho

Lucas Borges de Moraes<sup>1</sup>, Robson Berthelsen<sup>1</sup>, Adriano Fiorese<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação – (DCC)  
Universidade do Estado de Santa Catarina – (UDESC)  
Caixa Postal 631 – 89.219-710 – Joinville – SC – Brasil

{lucasborges1292, rosbson.brs}@gmail.com, adriano.fiorese@udesc.br

**Resumo.** *Este trabalho visa especificar um método lógico/matemático capaz de selecionar os mais adequados provedores de nuvem, baseado na análise e pontuação dos valores individuais dos indicadores de desempenho associados a cada provedor candidato. O método é estruturado em três etapas e é constituído por um algoritmo que pontua, ordena e seleciona diferentes provedores, com base na utilidade de seus indicadores.*

## 1. Introdução

A computação em nuvem mostrou uma nova visão de prestação de serviços aos seus clientes e tornou-se um paradigma diferenciado de hospedagem e distribuição de serviços computacionais via Internet. O sucesso desse paradigma motivou o surgimento de um grande número de novas empresas provedoras de computação em nuvem, tornando a tarefa de escolha e seleção da mais adequada um processo complexo. A medição da qualidade de um provedor pode ser feita pela medição sistemática da qualidade de cada um dos indicadores de desempenho ou PIs (*Performance Indicators*) individuais [Garg et al. 2013], resultando em um determinado valor ou *ranking*. O método proposto utiliza os valores de diferentes tipos de dados (numéricos ou categóricos) coletados de cada provedor e armazenados em diferentes PIs, para pontuar uma lista de distintos provedores de acordo com os requisitos demandados. Trabalhos relacionados sobre como ranquear e selecionar provedores com base em indicadores já foram desenvolvidos [Garg et al. 2013, Sundareswaran et al. 2012]. Contudo, o diferencial do método proposto nesse artigo é a sua alta generalidade, sendo capaz de lidar inclusive com indicadores qualitativos. Assim, o método é agnóstico quanto a quais PIs utilizar, ou seja, o usuário pode solicitar (expressão de entrada) qualquer PI e valor desejado.

## 2. O Método de Seleção Proposto

A Figura 1 apresenta uma visão geral do método de seleção de provedores a ser descrito nesta Seção. A entrada de dados (*Inputs*) corresponde a uma lista  $P$  com os  $n$  diferentes provedores candidatos, cada qual com  $M$  diferentes PIs (cujos valores são conhecidos) e uma expressão de entrada (gerada pelo usuário do método) contendo os  $m$  PIs de interesse (subconjunto dos PIs conhecidos) e o nível de prioridade de cada um. Essa lista inicial  $P$  será diminuída dos provedores incompatíveis, logo na primeira etapa do método, com base na expressão de entrada e terá  $n'$  elementos, com  $n \geq n'$ . Se  $n' = 0$ , nenhum provedor disponível é compatível, e assim interrompe-se o processo com uma mensagem

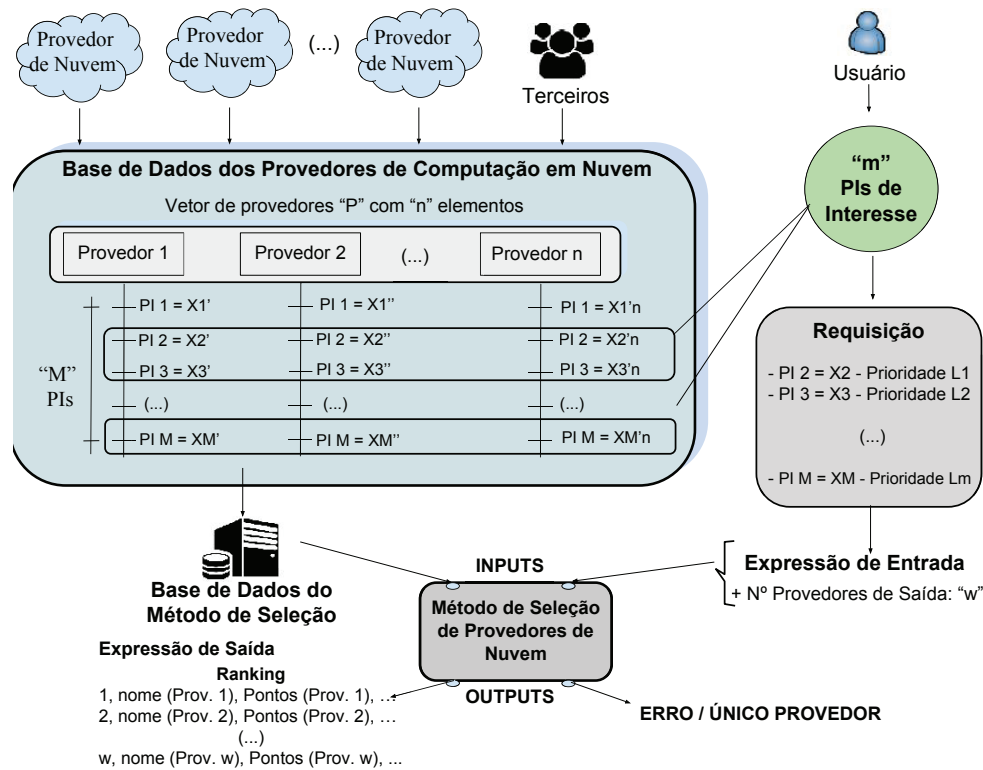


Figura 1. Método de seleção de provedores de nuvem baseado em PIs.

de erro; se  $n' = 1$  há um único provedor compatível, que é retornado; se  $n' > 1$  segue-se com as etapas seguintes.

A saída de dados (*Outputs*), exceto nas condições especiais mencionadas, será uma lista com os provedores mais bem pontuados pelo método. O banco de dados de candidatos provedores de nuvem e seus PIs pode ser alimentado indiretamente por meio de sites web como *Cloud Harmony* (<https://cloudharmony.com>) ou dos próprios provedores de nuvem. O método é composto por várias etapas, a seguir.

### 2.1. Etapa 1 – Eliminação dos Provedores Incompatíveis

A primeira etapa remove da lista inicial de provedores ( $P$ ), todo provedor incompatível ao usuário, gerando assim, uma nova lista  $P'$  com  $n'$  diferentes provedores. Cada PI pode ser classificado como essencial ou não essencial pelo usuário. Um PI essencial possui prioridade máxima. Ele indica que se não for atendido com aquele valor informado, o usuário não conseguirá atingir seus objetivos. Assim, um provedor é incompatível se ele não atender a todos os PIs essenciais do usuário. Os PIs não essenciais possuem diferentes níveis de prioridade, que podem variar entre os níveis “Alto” e “Baixo”. Os PIs podem ser classificados em duas classes [Jain 1991]: quantitativo (discreto ou contínuo) e qualitativo (ordenado ou não ordenado). Se o PI for quantitativo, há três classificações quanto ao seu comportamento [Jain 1991]: HB ou *Higher is Better*, LB ou *Lower is Better* e NB ou *Nominal is Best*. Assim, seja  $PI_j$  o  $j$ -ésimo PI quantitativo dos PIs de interesse, que armazena o valor  $x$ , e  $PI_j$  pertence ao provedor  $i$ , se  $PI_j$  atende ao valor  $y$  especificado pelo usuário, então conclui-se que:

$$\begin{cases} x \geq (y - t_{PI_j}) \text{ se } PI_j \in HB \\ x \leq (y + t_{PI_j}) \text{ se } PI_j \in LB \\ x = (y \pm t_{PI_j}) \text{ se } PI_j \in NB \end{cases} \quad (1)$$

Para os PIs qualitativos será criada uma nomenclatura que indicará se um PI qualitativo ordenado possui tolerâncias para valores abaixo e/ou acima daquele desejado: HT ou *Higher is Tolerable*, LT ou *Lower is Tolerable* e HLT ou *Higher and Lower is Tolerable*. Assim, dado um PI qualitativo  $PI_j$ , que armazena o valor  $x$  (categoria), e  $PI_j$  pertence ao provedor  $i$ , se  $PI_j$  atende ao valor  $y$  especificado pelo usuário, conclui-se que:

$$\left\{ \begin{array}{l} x = y \text{ se } PI_j \text{ é não ordenado OU } PI_j \in NB \\ x \geq y \text{ se } PI_j \text{ é ordenado E } PI_j \in HT \\ x \leq y \text{ se } PI_j \text{ é ordenado E } PI_j \in LT \\ PI_j \text{ é ordenado E } PI_j \in HLT \end{array} \right. \quad (2)$$

## 2.2. Etapa 2 – Pontuação dos PIs para cada Nível de Prioridade

Essa etapa pontua cada provedor individualmente, de acordo com a utilidade (benefício real) de cada um de seus PIs. Essa etapa recebe a lista  $P'$ , com os  $n'$  provedores restantes da etapa anterior. Cada provedor terá valores para os  $m$  PIs de interesse do usuário, que podem ser quantitativos ou qualitativos. Cada um desses PIs possui um nível de prioridade associado. A pontuação do nível de prioridade  $l$  ( $Pts_l$ ) é a média aritmética simples da pontuação individual de cada  $PI_j$  para esse mesmo nível de prioridade  $l$ , seja ele quantitativo ou qualitativo.

A pontuação de um PI quantitativo  $j$  de um provedor  $i$ , será 0, se seu valor numérico  $x$  não atende ao valor numérico  $y$  especificado pelo usuário. Caso o valor atenda, ele será pontuado proporcionalmente ao quão útil esse valor é em relação a todos os outros provedores disponíveis na lista de seleção. A função de avaliação de um PI quantitativo está representada na Eq. 3. Ela retorna sempre um valor real normalizado entre 0 e 1. O número  $X_{max}$  é o maior valor encontrado dentre todos os  $n'$  provedores na lista  $P'$  para aquele  $PI_j$ ; assim como  $X_{min}$  é o menor valor e  $PI_t$  a distância máxima tolerada do ponto ótimo ( $y$ ) para um PI do tipo NB.

$$Pqt(PI_j) = \left\{ \begin{array}{l} 0, \text{ se } valor(PI_j) \text{ não atende } y \\ 0.8 + 0.2 * \frac{valor(PI_j) - y}{X_{max} - y}, \text{ se } PI_j \in HB \\ 0.8 + 0.2 * \frac{y - valor(PI_j)}{y - X_{min}}, \text{ se } PI_j \in LB \\ 0.8 + 0.2 * \frac{PI_t - |y - valor(PI_j)|}{PI_t}, \text{ se } PI_j \in NB \end{array} \right. \quad (3)$$

Para os PI qualitativos ordenados, a pontuação depende da tolerância suportada e informada pelo usuário (HT, LT ou HLT) em relação ao valor (categoria) do PI ofertado pelo provedor. A pontuação de um PI qualitativo ordenado será sempre um valor real entre 0 e 1. A categoria desejável recebe pontuação máxima 1, as imediatamente vizinhas (superior e inferior) terão pontuação de 0.5 e os demais variam entre o intervalo aberto de 0 a 0.5 conforme a Eq. 4. A variável  $K_1$  é o número de categorias acima da categoria

desejada e  $K_2$  o número de categorias abaixo da mesma.

$$Pql(PI_j) = \begin{cases} 1, & \text{se } valor(PI_j) = valor(y) \\ 0.5 * \frac{K_1 - dist(valor(PI_j), y) + 1}{K_1}, & \text{se } PI_j \in HT \\ 0.5 * \frac{K_2 - dist(valor(PI_j), y) + 1}{K_2}, & \text{se } PI_j \in LT \\ 0.5 * \frac{K_1 + K_2 - dist(valor(PI_j), y) + 1}{(K_1 + K_2)}, & \text{se } PI_j \in HLT \\ 0, & \text{Caso contrário} \end{cases} \quad (4)$$

A Eq. 5 apresenta a pontuação por nível de prioridade, como a soma da pontuação dos PIs quantitativos e qualitativos que pertencem a aquele nível de prioridade  $l$ .

$$P_l(i) = \sum_{j=1}^m (Pqt(PI_j) + Pql(PI_j)) \quad [PI_j \in l] \quad (5)$$

### 2.3. Etapa 3 – Cálculo da Pontuação Final de cada Provedor

Por meio das duas etapas anteriores foram calculadas pontuações para cada um dos  $L$  diferentes níveis de prioridade, para cada provedor da lista  $P'$ . A consolidação dessas pontuações em um único valor corresponderá à pontuação do provedor em questão e será uma média aritmética ponderada, onde os coeficientes de ponderação (pesos) são diretamente proporcionais ao nível de prioridade que aquela pontuação se refere, de acordo com a Eq. 6.

$$P_{final}(i) = \sum_{l=1}^L (\alpha_l * P_l(i)) \quad (6)$$

## 3. Conclusão

Esse trabalho especificou um método auxiliar à tomada de decisão que pontua e ordena, visando a seleção, o mais apropriado provedor de nuvem, dentro um conjunto inicial de provedores, com base na requisição do usuário. A requisição deve apresentar os PIs de interesse, as preferências de valores para os mesmos e a prioridade de um sobre o outro. A pontuação final é um número real entre 0 e 1. Quanto mais perto de 1 mais adequado e preferível é o provedor em relação aos seus concorrentes para aquele usuário em questão. O método foi implementado em Java e submetido inicialmente a instâncias pequenas de testes (até 15 provedores com 10 PIs cada) e com dados hipotéticos. O método pontuou e selecionou adequadamente os provedores mais adequados a cada requisição gerada.

## Referências

- Garg, S. K., Versteeg, S., and Buyya, R. (2013). A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 29:1012–1023.
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, Littleton, Massachusetts.
- Sundareswaran, S., Squicciarini, A., and Lin, D. (2012). A brokerage-based approach for cloud service selection. In *2012 IEEE Fifth International Conference on Cloud Computing*, pages 558–565.