Seleção de Métricas Efetivas na Detecção de Anomalias em Sistemas Multi-camadas usando Correlação Parcial

Otto Julio Ahlert Pinno[1,2], Sand Luz Correa[1], Aldri Luiz dos Santos[2], Kleber Vieira Cardoso[1]

[1]Instituto de Informática (INF) – Universidade Federal de Goiás (UFG)

[2]Departamento de Informática – Universidade Federal do Paraná (UFPR)

{ottosilva,sand}@inf.ufg.br, aldri@inf.ufpr.br, kleber@inf.ufg.br

Abstract. *Large-scale data centers allow organizations to gain access to computer resources without incurring high costs in purchasing and maintaining IT infrastructure. In these environments, due to the large number of hardware and software involved, anomaly detection is difficult but essential for service provisioning. Computer systems hosted in data centers usually involve multiple layers and provide a large set of metrics for tracking their operation. The analysis of all available metrics generates drawbacks associated with communication, storage and processing. A more efficient way to support anomaly detection and minimize the cost of monitoring is to use stable statistical correlations among metrics that reflect the system state. We present the strategies PCTN, MST-PCTN and PCTN-MST, based on partial correlation, for selecting metrics to support anomaly detection in multi-tier systems. We evaluate the proposed strategies using an e-commerce, Web transaction benchmark. Results show that the PCTN-MST strategy allowed the construction of a monitoring network with 8% less metrics than that obtained with MST and achieved a fault coverage up to 10% larger.*

Resumo. *Centros de dados de larga escala permitem que organizações tenham acesso a recursos computacionais sem incorrer em alto custo de aquisição e manutenção de infraestrutura de TI. Nesses ambientes, devido aos inúmeros recursos de hardware e software envolvidos, a detecção de anomalias, embora difícil, se torna essencial à manutenção dos serviços aos usuários. Sistemas computacionais hospedados em centros de dados geralmente envolvem múltiplas camadas e disponibilizam um grande conjunto de métricas para coleta de dados sobre o seu funcionamento. A análise de todas as métricas disponíveis gera inconvenientes associados à comunicação, armazenamento e processamento dos dados. Uma forma mais eficiente de apoiar a detecção de anomalias e minimizar o custo do monitoramento é o emprego de correlações estatísticas estáveis entre métricas que refletem o estado do sistema. Neste trabalho, apresentamos as estratégias PCTN, MST-PCTN e PCTN-MST, baseadas em correlação parcial, para a seleção de métricas que apoiam a detecção de anomalias em sistemas multi-camadas. Avaliamos essas estratégias usando um benchmark de transações Web. Os resultados mostram que a estratégia PCTN-MST permitiu a construção de uma rede de monitoramento com* 8% *a menos de métricas que a estratégia MST e alcançou uma cobertura de falhas até* 10% *maior.*

1. Introdução

Cada vez mais organizações têm feito uso de infraestruturas de computação providas por centros de dados de larga escala. Esse modelo oferece flexibilidade pois permite que empresas tenham acesso a recursos sob demanda, sem a necessidade de incorrer em alto custo de aquisição e manutenção de infraestruturas de TI. No entanto, a operação e o gerenciamento de um centro de dados é uma tarefa complexa devido ao grande número de recursos de hardware e software que geralmente constituem esse ambiente [Aceto et al. 2013].

Tipicamente, centros de dados hospedam aplicações *Web*, tais como serviços de comércio eletrônico, redes sociais e serviços de transmissão de conteúdo multimídia. Essas aplicações são, em sua grande maioria, sistemas complexos compostos por várias camadas, como a de apresentação provida por um servidor HTTP (por exemplo o Apache), a de lógica da aplicação provida por um servidor de negócio (por exemplo o Tomcat) e a de persistência provida por um servidor de banco de dados (por exemplo o MySQL). Cada camada é um serviço independente e a sobreposição de eventos como *locks* de dados, variabilidade de tempo de serviço de uma operação, contenção de memória e concorrência, nas diferentes camadas, pode levar a padrões de execução anormais ou até mesmo a interrupção completa do serviço oferecido para o usuário [Mi et al. 2010].

Apesar dessas dificuldades, é esperado que as aplicações *Web* apresentem alta disponibilidade, confiabilidade e responsividade, sob pena de causarem perdas econômicas expressivas [Wang et al. 2013]. Estima-se que a parada ou interrupção de *Web sites* empresariais tem um custo médio de US$ 21 mil por hora para suas organizações proprietárias [Simic 2014]. O alto custo de parada dos serviços *online* atuais aliado à complexidade de grandes aplicações *Web* levam à necessidade de ferramentas que possam auxiliar os administradores a rastrear o comportamento desses sistemas e detectar padrões de execução anormais. Para auxiliar o desenvolvimento dessas ferramentas, sistemas de computação que operam em centros de dados disponibilizam uma grande quantidade de dados sobre o funcionamento de diversos componentes, incluindo dados de arquivos de *log* dos componentes das aplicações, métricas relacionadas às tecnologias de *middleware* usadas nos sistemas, eventos de auditoria, além de estatísticas relacionadas a métricas do sistema operacional e tráfego da rede. No entanto, a coleta de todos esses dados afeta não apenas o desempenho dos sistemas mas também gera inconvenientes associados à comunicação, armazenamento e processamento dos dados.

Uma abordagem mais eficiente para apoiar a detecção de anomalias em sistemas que seguem uma arquitetura multi-camadas composta por vários serviços, como as aplicações *Web*, consiste no uso de correlações estatísticas estáveis entre métricas que refletem o estado do sistema [Jiang et al. 2006, Jiang et al. 2009, Munawar et al. 2009]. Nessa abordagem, correlações entre métricas são capturadas através de modelos matemáticos que descrevem o comportamento de uma métrica em função do estado de outra. Como apenas correlações estáveis são consideradas, é esperado que as relações descritas pelos modelos mantenham-se válidas enquanto o sistema operar livre de falhas. No entanto, essas relações serão violadas quando ocorrerem falhas no sistema. A seleção de correlações estáveis tem também o efeito de filtrar um subconjunto de métricas dentre o conjunto disponível. Dessa forma, uma vez encontradas as correlações estáveis, apenas as métricas envolvidas nessas correlações passam a ser coletadas e monitoradas periodicamente, reduzindo o custo de monitoramento. Uma dificuldade, no entanto,

consiste em encontrar tais correlações. Existem algumas soluções para esse problema, tais como a validação de modelos de forma iterativa [Jiang et al. 2006] ou a seleção de correlações estáveis usando coeficiente de Pearson [Magalhaes and Silva 2010]. Porém essas soluções apresentam restrições, discutidas na Seção 2.

Neste trabalho, investigamos uma nova abordagem baseada em correlação parcial [Baba et al. 2004], a qual descreve o quanto o relacionamento entre duas variáveis (métricas) é resultado de suas correlações com uma variável intermediária. Essa informação permite excluir relacionamentos que são resultados de correlações indiretas, obtendo um conjunto mínimo de correlações estáveis. Neste trabalho, apresentamos as estratégias PCTN, MST-PCTN e PCTN-MST, baseadas em correlação parcial, para a seleção de métricas que apoiam a detecção de anomalias em sistemas multi-camadas. Avaliamos a efetividade dessas estratégias através de um conjunto de experimentos detalhados usando o TPC-W [Menasce 2002], um *benchmark* de transações *Web* para sistemas de comércio eletrônico. Os resultados mostram que a estratégia PCTN-MST permitiu a construção de uma rede de monitoramento com 8% a menos de métricas que aquela obtida usando MST, a melhor estratégia existente atualmente para esse problema, e alcançou uma cobertura de falhas até 10% maior.

Este trabalho está organizado da seguinte forma. A Seção 2 apresenta alguns fundamentos e trabalhos relacionados. A Seção 3 descreve as estratégias propostas. A Seção 4 apresenta a avaliação experimental e os resultados encontrados. Finalmente, a Seção 5 apresenta as conclusões e trabalhos futuros.

2. Fundamentos e Trabalhos Relacionados

Ferramentas comerciais como o SmartCloud [IBM 2014] e o System Center Operations Manager [Microsoft 2013] frequentemente adotam soluções baseadas em regras para detectar anomalias em centros de dados. Nessas soluções, os administradores configuram manualmente regras que estabelecem limiares para algumas métricas a serem monitoradas. Quando o valor de uma métrica extrapola seu limiar, uma anomalia é detectada e um alarme é emitido para o administrador. No entanto, a observação individualizada de métricas não é suficiente para descrever o comportamento de um sistema complexo [Jiang et al. 2006, Chen et al. 2010, Wang et al. 2014].

Uma forma mais eficaz consiste em observar também possíveis correlações entre métricas, pois essas correlações podem capturar a dinâmica do sistema à medida que as requisições de usuários fluem através dos componentes da aplicação. Considere, por exemplo, uma aplicação *Web*. Se existe uma correlação direta entre o número de requisições HTTP para o servidor *Web* e o número de requisições SQL para o servidor de banco de dados, então podemos esperar que um aumento de requisições no primeiro servidor leve a um aumento de requisições no segundo. Se esse relacionamento é estável, ou seja, ele se mantém durante a operação normal do sistema, podemos tomá-lo como uma invariante. Quando uma falha ocorre, a dinâmica do sistema é afetada e algumas invariantes são violadas. Portanto, podemos detectar falhas num sistema monitorando suas invariantes, ou seja, suas correlações estáveis. Esta forma de descrever o comportamento de um sistema é denominado *monitoramento baseado em correlação* [Jiang et al. 2009].

O monitoramento baseado em correlação apresenta diversas vantagens. Essa abordagem não exige nenhuma informação de entrada por parte do administrador, pois ne-

nhum conhecimento prévio da estrutura do sistema é requerido. Além disso, essa abordagem é genérica, podendo ser aplicada em qualquer sistema com monitoramento de dados. Por fim, a seleção de correlações estáveis reduz o número de métricas monitoradas, reduzindo a sobrecarga imposta pelo monitoramento. No entanto, o desafio nessa abordagem é encontrar as correlações verdadeiramente estáveis.

Para resolver esse problema, os trabalhos existentes em monitoramento baseado em correlação utilizam diferentes abordagens. [Jiang et al. 2006] usam regressão linear para capturar o relacionamento entre pares de métricas monitoradas por um sistema *Web*. Para determinar quais relações lineares são estáveis, uma vez que os modelos são gerados para todas as combinações possíveis envolvendo duas métricas, os autores usam um método iterativo e estabelecem um fator de confiança, o qual é medido periodicamente a partir do número de previsões corretas emitidas pelos modelos. Quando o fator de confiança de um modelo cai abaixo de um limiar, ele é removido da solução. Um problema dessa abordagem é sua convergência lenta para o conjunto mínimo de métricas.

Uma alternativa ao método iterativo proposto por [Jiang et al. 2006] consiste em calcular o coeficiente de Pearson para quantificar a intensidade do relacionamento entre duas métricas. Essa abordagem foi utilizada por [Magalhaes and Silva 2010] para detectar problemas de desempenho em aplicações *Web*. Dadas n observações de duas variáveis X e Y, o coeficiente de Pearson entre essas variáveis é denotado pela Equação 1 e representa o quanto as duas variáveis mudam de maneira similar (covariância) relativamente a suas dispersões (desvios padrões):

$$\rho(X,Y) = \frac{cov(X,Y)}{s_X s_Y} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}, \quad (1)$$

onde X_i e Y_i são observações das variáveis X e Y, respectivamente, e $\bar{X}$ e $\bar{Y}$ são as médias das n observações de X e Y, respectivamente. ρ assume valores no intervalo $[-1, 1]$. Em geral, assume-se que a correlação é forte se o valor absoluto de ρ está no intervalo $[0.5, 1]$ [Cohen 1988]. A definição de correlação forte pode ser usada para encontrar as correlações estáveis. Nesse sentido, consideramos estáveis apenas as correlações fortes.

Uma limitação do coeficiente de Pearson é que ele não provê nenhuma informação sobre a existência de uma terceira métrica condicionando o relacionamento observado entre duas outras métricas. Como consequência, o coeficiente de Pearson pode retornar um grande número de correlações indiretas. Sejam X, Y e Z três métricas onde identificamos uma forte correlação entre os pares (X, Y), (X, Z) e (Y, Z). Dizemos que a correlação entre o par (X, Y) é indireta, se ela é resultado das correlações individuais de X e Y com Z. Correlações indiretas representam informações redundantes e, portanto, podem ser eliminadas do monitoramento.

[Munawar et al. 2009] apresentam uma abordagem baseada em árvore geradora mínima (*Minimum Spanning Tree* - MST) para eliminar correlações indiretas. Após modelar as relações entre todos os pares possíveis de métricas, os autores utilizam o coeficiente de Pearson para selecionar as correlações estáveis. As correlações selecionadas são usadas para construir uma rede onde métricas são mapeadas em vértices e correlações em arestas. As correlações mais importantes dessa rede são então capturadas usando a MST.

Diferentemente de [Munawar et al. 2009], neste trabalho, usamos o conceito de *correlação parcial* para eliminar as correlações indiretas e encontrar um conjunto mínimo de correlações estáveis que garanta cobertura de falhas. A correlação parcial permite determinar se o relacionamento entre um par de métricas (X, Y) é condicionado por outra métrica Z. Essa verificação é realizada removendo a influência de Z e recalculando a correlação entre o par (X, Y). Se o resultado da correlação parcial for significativamente menor que o da correlação original, então a correlação original existia, em sua maior parte, devido à correlação de X e Y com Z. A correlação parcial é expressa pelo coeficiente de correlação parcial. Dadas duas variáveis X e Y e uma variável condicionante Z, o coeficiente de correlação parcial pode ser expresso em termos dos coeficientes de Pearson $\rho(X, Y)$, $\rho(X, Z)$ e $\rho(Y, Z)$ como mostrado na Equação 2:

$$\rho(X, Y : Z) = \frac{\rho(X, Y) - \rho(X, Z)\rho(Y, Z)}{\sqrt{[1 - \rho^2(X, Z)][1 - \rho^2(Y, Z)]}}. \quad (2)$$

A correlação parcial tem sido empregada em áreas como Biologia e Economia para estudar o comportamento de sistemas complexos [De La Fuente et al. 2004, Kenett et al. 2010]. Nesses trabalhos, a correlação parcial é aplicada para eliminar muitas correlações que não representam relacionamentos reais. Em redes ou sistemas complexos, isso é útil pois reduz o conjunto de correlações a ser analisado. Até onde sabemos, este é o primeiro trabalho que usa correlação parcial para estudar redes de métricas monitoradas em um sistema computacional.

Finalmente, é importante mencionar que alguns trabalhos utilizam busca baseada em projeção estatística para selecionar um conjunto de métricas para o monitoramento, dado um critério de interesse. PCA (*Principal Component Analysis*) é uma dessas abordagens, onde o critério é a variância das amostras. No entanto, essa técnica não é aplicável para sistemas multi-camadas, pois a carga de trabalho nesses sistemas é variável e nem sempre apresenta um comportamento regular [Peiris et al. 2014].

3. Monitoramento Baseado em Correlação Usando Correlação Parcial

Antes de introduzirmos as estratégias baseadas em correlação parcial que propusemos para selecionar correlações estáveis, descrevemos, na Subseção 3.1, o arcabouço que adotamos para detecção de anomalias em sistemas multi-camadas no contexto de monitoramento baseado em correlação. As estratégias são apresentadas na Subseção 3.2.

3.1. Modelagem do Comportamento do Sistema e Detecção de Anomalias

Usando dados de métricas coletadas de um sistema multi-camadas operando livre de falhas, dividimos esses dados em dois conjuntos: um de treinamento e outro de teste. Em seguida, implementamos as 4 atividades descritas a seguir e ilustradas na Figura 1.

1. **Identificação de correlações estáveis**: para cada par de métricas X e Y monitoradas, consideramos que existe uma correlação estável entre X e Y se, para qualquer métrica Z diferente de X e de Y, a correlação parcial $\rho(X, Y : Z)$ é maior que um certo limiar. A identificação de correlações estáveis é feita sobre o conjunto de treinamento.

2. **Construção de modelos**: para cada par de métricas X e Y para o qual o passo anterior identificou a existência de uma correlação estável, utilizamos uma técnica de regressão

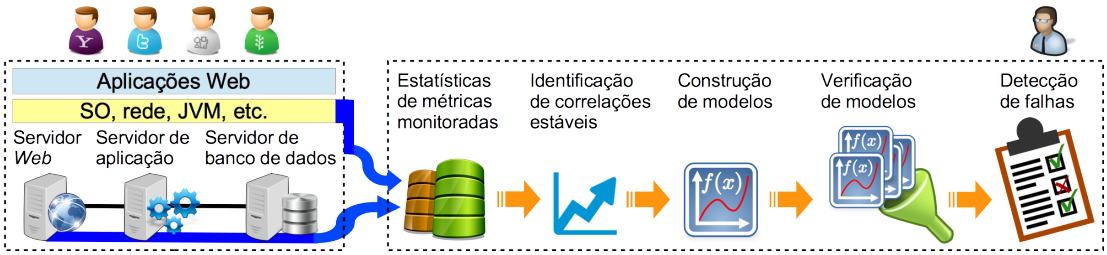

Figura 1. Visão geral do mecanismo de monitoramento.

linear sobre o conjunto de treinamento e construímos um modelo matemático $Y = f(X)$ para descrever o relacionamento entre esse par de métricas.

3. **Verificação de modelos**: para cada modelo matemático construído no passo anterior, verificamos mais uma vez a sua estabilidade, usando o conjunto de teste. Para realizar essa verificação, para cada modelo, construímos um *limite de aceitação de resíduo*. Sejam X_i e Y_i os i-ésimos valores observados para as métricas X e Y, respectivamente, no conjunto de teste. Seja $\hat{Y}_i$, o valor previsto pelo modelo quando X_i é fornecido como entrada. Definimos o resíduo como sendo a diferença entre o valor realmente observado para Y em um instante de tempo e o valor estimado pelo modelo, ou seja, $R_i = Y_i - \hat{Y}_i$.

Para determinar os limiares superior e inferior do limite de aceitação de um modelo, usamos a mesma abordagem empregada em [Jiang et al. 2006]. Seja R_i, $i = 1...n$, os resíduos gerados pelo modelo. Essa abordagem consiste em encontrar um valor $\hat{r}$ que é maior que $99,5\%$ dos resíduos observados e amplificar o resultado em 10%, como descrito na Equação 3. Os valores positivos e negativos do limiar resultante (τ) são usados, respectivamente, como limiares superior e inferior do limite de aceitação do modelo.

$$\tau = 1,1 \times arg_{\hat{r}}\{prob(|R_i| \leq \hat{r}) = 0.995\} \tag{3}$$

A verificação de um modelo é realizada fornecendo os valores X_i, do conjunto de teste, como entrada e observando os resíduos R_i gerados. É considerado estável o modelo em que até uma determinada porcentagem de observações de resíduos estão fora do limite de aceitação. Chamamos esse limiar de *Porcentagem de Resíduos Discrepantes Permitidos* - PRDP.

4. **Detecção de falhas**: a detecção de anomalias ocorre através do monitoramento dos modelos que passaram no teste de verificação. Periodicamente, esses modelos são alimentados com novas observações de métricas e os resíduos são calculados. Se uma porcentagem de resíduos maior que PRDP viola o limite de aceitação, o modelo correspondente é violado, indicando a ocorrência de uma anomalia ou falha.

É importante mencionar que as métricas monitoradas ficam restritas àquelas usadas pelos modelos que passaram no teste de verificação, reduzindo o custo de monitoramento. Esses modelos foram selecionados a partir da identificação das correlações estáveis. Essa seleção prévia acelera a convergência para um conjunto pequeno de modelos e, consequentemente, de métricas monitoradas. A seguir, descrevemos três estratégias que propusemos para identificar correlações estáveis em um sistema multi-camadas.

3.2. Estratégias para Seleção de Correlações Estáveis

Na Equação 2, um valor alto de $\rho(X, Y : Z)$ indica que Z exerce pouca ou nenhuma influência na correlação entre o par de métricas (X, Y). Isso ocorre porque o coeficiente de correlação parcial assume um valor alto apenas se $\rho(X, Y)$ for muito maior que $\rho(X, Z).\rho(Y, Z)$. Por outro lado, um valor pequeno de $\rho(X, Y : Z)$ pode indicar duas situações: (1) Z tem uma forte influência na correlação entre o par (X, Y), ou seja, $\rho(X, Y) \sim \rho(X, Z).\rho(Y, Z)$, (2) a correlação entre as três métricas é pequena, ou seja, os coeficientes de Pearson $\rho(X, Y)$, $\rho(X, Z)$ e $\rho(Y, Z)$ são pequenos. Para diferenciar entre essas duas situações, frequentemente, a correlação parcial é calculada através da influência de correlação [Kenett et al. 2010], mostrada na Equação 4:

$$d(X, Y : Z) \equiv \rho(X, Y) - \rho(X, Y : Z). \tag{4}$$

A influência de correlação quantifica a influência de Z no relacionamento do par de métricas (X, Y). Esse valor é alto apenas quando uma parte significativa da correlação entre X e Y é explicada em função da métrica Z.

Nossa primeira estratégia para identificar as correlações estáveis em um sistema multi-camadas consiste em modelar uma rede de monitoramento usando apenas a influência de correlação. Essa estratégia é denominada *Partial Correlation Threshold Network* - PCTN por utilizar o algoritmo PCTN proposto por [Kenett et al. 2010].

O algoritmo PCTN cria uma rede composta pelas influências de correlação $d(X, Y : Z)$ com valores maiores que um certo limiar. Essa rede é representada por um grafo, onde os vértices são as métricas monitoradas e as arestas correspondem às correlações entre as métricas. Para cada combinação de três métricas X, Y e Z, acrescentamos uma aresta entre Z e X e outra aresta entre Z e Y, indicando a influência de Z sobre X e Y, se, e somente se, elas ainda não existem na rede e a Equação 5 for satisfeita:

$$d(X, Y : Z) \geq \langle d(X, Y : Z)\rangle_Z + k\sigma_Z(d(X, Y : Z)), \tag{5}$$

onde $\langle d(X, Y : Z)\rangle_Z$ e $\sigma_Z(d(X, Y : Z))$ são, respectivamente, a média e o desvio padrão determinados com respeito à métrica condicionante Z. O parâmetro k, denominado limiar de influência, define a intensidade de atuação do limiar.

A topologia da rede a ser construída na PCTN depende fortemente de k. Valores altos podem levar a um grafo pouco conexo, o que prejudica a capacidade de detecção de anomalias de um mecanismo. Isso ocorre porque cada aresta representa uma correlação e cada correlação dará origem a um modelo que será monitorado. Se existem poucas arestas no grafo, o mecanismo irá monitorar poucos modelos e alguns relacionamentos importantes podem não ser representados. Por outro lado, valores muito pequenos podem aproximar a topologia da rede de um grafo completo, diminuindo a capacidade de filtragem de correlações da rede. Assim como [Kenett et al. 2010], utilizamos uma abordagem empírica para a escolha de k. Entretanto, nossa abordagem é diferente da original, como descreveremos na Subseção 4.2.1.

As duas outras estratégias propostas neste trabalho são combinações das estratégias PCTN e MST. A estratégia MST-PCTN identifica as correlações estáveis da seguinte forma. Dado o conjunto contendo todas as métricas monitoradas, calculamos

a correlação (de Pearson) para todos os pares possíveis de métricas. O conjunto de correlações é reduzido, inicialmente, construindo uma rede de monitoramento usando a estratégia MST proposta por [Munawar et al. 2009]. Em seguida, aplicamos a estratégia PCTN sobre a rede resultante, gerando outra rede. As arestas da rede final representam as correlações estáveis.

A estratégia PCTN-MST segue a mesma lógica, porém as estratégias são aplicadas na ordem inversa. Dado o conjunto contendo todas as métricas monitoradas, calculamos a correlação (parcial) para cada combinação de três métricas usando a estratégia PCTN. Em seguida, aplicamos a estratégia MST sobre a rede formada pela PCTN, gerando outra rede. As arestas da rede final representam as correlações estáveis.

4. Avaliação de Desempenho

Para realizar a avaliação das estratégias propostas, preparamos um ambiente de testes com uma aplicação *Web* multi-camadas, o qual é descrito em detalhe na Subseção 4.1. Além dessas estratégias, implementamos e avaliamos outras abordagens, inclusive a MST proposta por [Munawar et al. 2009], conforme mostraremos na Subseção 4.2.

4.1. Infraestrutura de Testes e Carga de Trabalho

A Figura 2 ilustra o ambiente de testes implantado para a realização dos experimentos. Como pode ser visto, o ambiente é uma arquitetura de três camadas, formada por: 1) camada de dados, 2) camada de negócio ou lógica e 3) camada de apresentação. A camada de dados (`Database Server`) é implementada pelo servidor MySQL, enquanto a camada de negócio (`Front Server`) é baseada no servidor de aplicação Apache Tomcat. No `Front Server`, também está instalada a lógica da aplicação *Web* que se baseou no núcleo do TPC-W. O TPC-W é um *benchmark* clássico que emula comportamentos comuns a *sites* de comércio eletrônico [Menasce 2002]. Neste trabalho, utilizamos uma implementação em Java de código aberto do TPC-W [Bezenek et al. 2014]. A camada de apresentação (`Client`) também é implementada a partir do TPC-W através de um módulo que emula clientes da aplicação *Web*. Os equipamentos `Front Server` e `Database Server` também executam a ferramenta collectd [Forster 2014] para realizar a coleta de estatísticas das métricas monitoradas. A Tabela 1 resume as principais características dos equipamentos. Todos os equipamentos utilizam o sistema operacional Linux com a distribuição Ubuntu 12.04 LTS.

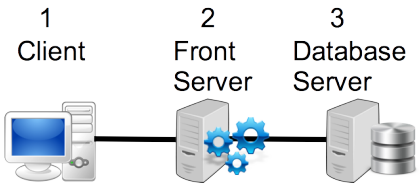

Figura 2. Arquitetura do ambiente de testes.

Equip.	Processador	Memória	Rede
1	Atom @ 1,66 GHz	1 GB	1 Gbps
2	Core i5 @ 3,20 GHz	4 GB	1 Gbps
3	Core i5 @ 3,20 GHz	4 GB	1 Gbps

Tabela 1. Especificação básica dos equipamentos.

Tipicamente, a carga de trabalho de uma aplicação *Web*, como comércio eletrônico e *sites* empresariais, pode ser descrita adequadamente no nível de sessão [Zhang et al. 2007]. Uma sessão consiste em uma sequência de requisições individuais e consecutivas. Todas as requisições individuais são necessárias para completar uma transação de mais alto nível. Portanto, uma medida de desempenho comumente

usada em aplicações *Web* é o número de sessões concorrentes que o sistema pode suportar sem violar um tempo de resposta transacional máximo estabelecido previamente.

De acordo com a especificação do TPC-W, o número de sessões concorrentes, também chamadas de navegadores emulados (*emulated browsers* – EB), é constante durante todo o experimento. Para cada *EB*, o TPC-W define, de forma estatística, o tamanho da sessão do cliente, o tempo de pensar (*think time*) do usuário e as consultas que são geradas pela sessão. O tamanho médio das sessões é de 15 minutos e sempre que uma sessão termina, uma nova é criada. Além disso, o tamanho do banco de dados é determinado pelo número de itens disponíveis para compra e também pelo número de clientes.

O TPC-W define 14 transações e 3 grupos de distribuição de acesso entre essas transações, a saber: *Browsing Mix*, *Shopping Mix* e *Ordering Mix*. A título de ilustração, a Tabela 2 lista 6 das transações e as porcentagens médias de requisições destinadas às transações de cada grupo de distribuição. Em nossa avaliação, optamos pelo *Browsing Mix* para a geração da carga, uma vez que esse grupo representa um padrão de acesso fortemente baseado em leitura. Esse tende a ser o padrão mais comum em ambientes reais, pois grande parte das transações realizadas em *sites* de comércio eletrônico envolvem pesquisas que não resultam efetivamente em pedido ou compra.

Interação Web	**Browsing Mix**	**Shopping Mix**	**Ordering Mix**
Pesquisa	**95%**	**80%**	**50%**
Home	29,00%	16,00%	9,12%
Product Detail	21,00%	17,00%	12,35%
Search Request	12,00%	20,00%	14,54%
Pedidos	**5%**	**20%**	**50%**
Shopping Cart	2,00%	11,60%	13,53%
Customer Reg.	0,82%	3,00%	12,86%
Buy Request	0,75%	2,60%	12,73%

Tabela 2. Distribuição das requisições entre as transações do TPC-W.

Inicialmente, avaliamos o sistema livre de falhas, pois o intuito era construir e treinar os modelos que descreveriam a relação entre os pares de métricas. Através de experimentos, verificamos que o sistema é capaz de tratar até 300 EBs sem apresentar atrasos sensíveis na coleta de estatísticas. Realizamos mais de 50 experimentos de 30 minutos, nos quais eram monitoradas 396 métricas. Essas métricas incluem estatísticas sobre utilização de CPU, memória, disco e rede, além de métricas do sistema operacional e da máquina virtual Java (JVM). A coleta de estatísticas é realizada ao final de cada janela de monitoramento que tem tamanho fixo de 5 segundos. Esse intervalo ofereceu um compromisso adequado entre a quantidade de amostras e o impacto no consumo de CPU, o qual se mostrou negligenciável, permanecendo abaixo de 0,5%.

4.2. Resultados

Organizamos a apresentação dos resultados em três partes, as quais são descritas nas subseções que seguem: 1) quantidade de correlações e métricas no sistema livre de falhas, 2) estabilidade das correlações em função do percentual tolerado de resíduos discrepantes e 3) desempenho em termos de percentual de falhas detectadas e taxa de falso-positivos.

4.2.1. Correlações e Métricas

A partir da avaliação do sistema livre de falhas, formamos uma base de dados de treinamento usada para criar as redes de monitoramento. Essas redes foram criadas aplicando as estratégias PCTN, MST-PCTN e PCTN-MST. Além dessas redes, criamos também outras, a partir da mesma base de dados, aplicando as estratégias de Pearson e MST. A seguir, apresentamos o processo de criação de cada rede e a quantidade de correlações e métricas resultante em cada uma. É importante mencionar que essas redes foram criadas como saída da atividade 1 descrita na Seção 3.1.

Rede de Pearson

A partir da base de dados gerada pela carga *Browsing Mix*, calculamos o coeficiente de Pearson para cada par de métricas monitoradas. Em seguida, selecionamos todos os pares que apresentaram correlação forte, ou seja $\rho > 0,5$. Do total de 78.210 correlações que abrangiam as 396 métricas, foram identificadas como fortes **1.773 correlações** que envolviam **192 métricas**. Ou seja, o monitoramento dessa rede ainda possui custo alto.

PCTN

A partir da base de dados gerada pela carga *Browsing Mix*, calculamos a influência de correlação $d(X, Y : Z)$ para cada combinação de métricas X, Y e Z, conforme a Equação 4. A seguir, construímos o grafo da rede, acrescentando uma aresta entre Z e X e entre Z e Y, em todas as combinações em que a influência de correlação satisfazia a Equação 5. Conforme descrito previamente, essa expressão exige a definição de um limiar de influência k. O valor adequado de k deve ser suficientemente alto para reduzir de maneira significativa o número de arestas, mas também deve ser baixo o bastante para manter um componente conexo com grau razoavelmente alto na rede.

[Kenett et al. 2010] adotaram uma abordagem experimental para encontrar um k adequado, o qual consiste em variar o valor desse parâmetro de forma iterativa e avaliar, para cada rede formada, a soma dos pesos das arestas da rede e o grau do vértice mais conexo. O procedimento adotado para dar pesos às arestas é: para cada aresta $Z \rightarrow X$, o peso dessa aresta é igual ao número de variáveis Y que satisfazem a Equação 5. Seguindo uma estratégia similar, variamos o valor de k e analisamos as duas informações apresentadas por [Kenett et al. 2010] sobre as redes formadas, além de verificarmos a quantidade de influências retornadas pela PCTN.

Diferentemente de [Kenett et al. 2010], não observamos uma redução abrupta no somatório dos pesos das arestas ao mesmo tempo em que o grau do vértice mais conexo se mantinha relativamente estável. Em seu estudo, [Kenett et al. 2010] mostraram que algumas ações do mercado financeiro possuem alta influência sobre uma grande parte de outras ações. No entanto, no sistema computacional investigado, observamos que as influências tendem a ser mais distribuídas entre suas métricas. Por essa razão, tivemos que adequar essa metodologia para escolha de k e o fizemos da maneira descrita a seguir. Variamos iterativamente o valor desse parâmetro até que o número máximo de correlações retornadas pela PCTN fosse igual ao número de correlações existentes na rede de Pearson. Como duas correlações podem ser originadas de cada influência retornada pela Equação 5, a saber, uma correlacionando Z e X e a outra Z e Y, o número de influência retornada deve ser igual à metade do número de correlações na rede de Pearson. Para a carga

analisada, o número de influências definido foi alcançado com $k = 17,5$. Usando esse valor como limiar de influência na Equação 5, obtivemos uma rede com **396 correlações** e **177 métricas**. Logo, a PCTN apresenta uma demanda de monitoramento sensivelmente menor que a estratégia baseada em Pearson.

Rede MST

Conforme proposto por [Munawar et al. 2009], uma MST é construída para selecionar as métricas mais importantes em uma rede de Pearson. Dado um conjunto de métricas monitoradas, é inicialmente construída a rede de Pearson para selecionar as correlações fortes. Em seguida, cada aresta da rede é rotulada com um peso igual a $1 - \rho(X, Y)$, onde $\rho(X, Y)$ é o valor do coeficiente de Pearson da correlação representada pela aresta. Por fim, a MST é calculada para a rede rotulada. As arestas da MST representam as correlações identificadas como estáveis. A MST obtida possui **183 correlações** envolvendo **192 métricas**, isto é, a rede MST possui menos correlações que a PCTN, embora envolva um número maior de métricas.

MST-PCTN e PCTN-MST

Além de avaliar o desempenho da PCTN em relação à MST, analisamos o comportamento de redes formadas pela combinação das duas, a saber: MST-PCTN e PCTN-MST. A MST-PCTN é formada aplicando o filtro baseado em correlação parcial sobre as métricas selecionadas pela MST. Para tanto, definimos $k = 15,13$ de maneira que a PCTN retornasse uma quantidade de influências de correlação igual à metade da quantidade existente na MST, ou seja, a mesma estratégia usada com a rede Pearson. Em termos de custo de monitoramento, essa é a melhor abordagem, pois obtemos uma rede com apenas **62 correlações** e **42 métricas**. No entanto, conforme mostraremos posteriormente, essa rede apresentou baixa capacidade de detecção de falhas.

Inicialmente, a PCTN-MST é formada através da geração da PCTN a partir da base de dados e pela aplicação do limiar de influência $k = 17,5$. Em seguida, o filtro baseado em MST é aplicado sobre a rede formada pela PCTN. A rede resultante apresentou **174 correlações** e **177 métricas**. Logo, o número de métricas e também de correlações da PCTN-MST é menor que os da MST. Além disso, a PCTN-MST apresenta um compromisso melhor entre a quantidade de falhas detectadas e o número de falso-positivos, conforme apresentaremos na última parte da avaliação.

4.2.2. Estabilidade das Correlações

Apesar dos modelos matemáticos gerados a partir das redes de monitoramento terem sido criados de correlações identificadas como estáveis, é importante avaliar a estabilidade desses modelos. Um modelo é considerado estável se as relações invariantes persistem em diferentes execuções do sistema livre de falhas. Para avaliar a estabilidade das correlações geradas, submetemos os modelos a testes livres de falhas e observamos a quantidade que passa nos testes. Nessa avaliação, realizamos 20 testes para cada valor de PRDP, o qual varia de 0% a 90%. Todas as redes obtiveram seus melhores resultados dentro do intervalo de PRDP entre 1% e 3%, portanto, nos restringiremos a essa faixa de valores.

A Tabela 3 apresenta o número de modelos aceitos e a quantidade de métricas referenciadas pelos mesmos. É possível observar que o número de modelos, de todas as

redes, tende a aumentar de maneira significativa à medida que o PRDP aumenta. Por outro lado, a variação do PRDP tem pouco impacto na quantidade de métricas monitoradas. Além disso, com exceção da MST-PCTN, todas as redes possuem uma quantidade de métricas na mesma ordem de grandeza, ilustrando a dificuldade em encontrar soluções para reduzir o custo de monitoramento.

Rede de monitoramento	Correlações			Métricas		
	PRDP			PRDP		
	1%	2%	3%	1%	2%	3%
Pearson	340	735	900	182	184	184
PCTN	123	256	301	159	161	162
MST	55	117	135	170	175	175
MST-PCTN	16	33	44	40	41	41
PCTN-MST	48	112	129	158	160	161

Tabela 3. Quantidade de correlações e métricas após os testes de estabilidade.

4.2.3. Detecção de Falhas e Falso-Positivos

Para avaliar a capacidade de detecção de falhas dos modelos selecionados pelas redes de monitoramento, escolhemos algumas falhas típicas de sistemas computacionais [Wang et al. 2014] e as injetamos em nosso ambiente de testes. As falhas injetadas podem ser classificadas em: falha de software ou de programação, falha de desempenho, falha de configuração e falha de comunicação. Realizamos 31 experimentos com duração de 30 minutos cada. Em cada experimento, injetamos uma única falha, aproximadamente na metade do tempo do teste. A única exceção é a falha de configuração, a qual foi inserida logo no início do experimento. Para avaliar os falso-positivos, realizamos 31 experimentos livres de falhas. Conforme comentado previamente, os resultados se concentram na faixa de PRDP com os melhores desempenhos, isto é, entre 1% e 3%.

As Figuras 3 e 4 apresentam, respectivamente, o percentual de falhas detectadas e a taxa de falso-positivos das redes avaliadas. A rede de Pearson é capaz de detectar 100% das falhas, porém possui uma taxa de falso-positivos muito alta, próxima a 84% em sua melhor parametrização de PRDP. Além disso, é importante lembrar que a rede de Pearson possui o maior número de métricas monitoradas (184). Em outro extremo, está a MST-PCTN que, apesar de ser muito eficiente em termos de métricas monitoradas (40) e apresentar a menor taxa de falso-positivos, também exibe um percentual de detecção de falhas muito baixo, em torno de 20%. As outras redes operam fora desses extremos e há compromissos a serem analisados, conforme discutimos a seguir.

O melhor resultado da MST é uma detecção de falhas em torno de 70% e uma taxa de falso-positivos ligeiramente inferior a 50%. A PCTN detectou aproximadamente 84% de falhas e um pouco menos de 50% de taxa de falso-positivos. Ou seja, um desempenho superior à MST, o qual se torna ainda mais relevante por vir a um menor custo de monitoramento, uma vez que são usadas 162 métricas pela PCTN contra 184 da MST. Por fim, temos a PCTN-MST que tem praticamente a mesma quantidade de métricas que a PCTN e apresenta uma elevação no percentual de falhas detectadas (80%). No entanto, a PCTN-MST exibe uma taxa menor de falso-positivos (40%). Assim, a PCTN-MST alcança o

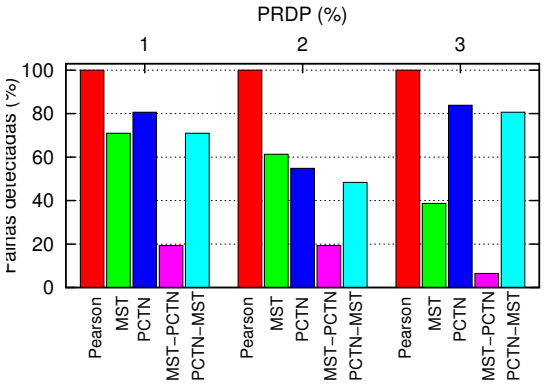

Figura 3. Porcentagem de falhas detectadas em função da PRDP.

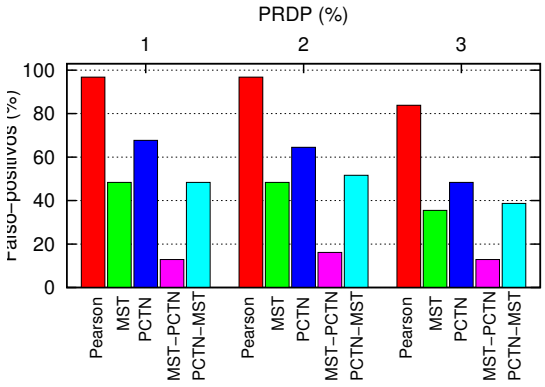

Figura 4. Taxa de falso-positivos em função da PRDP.

melhor compromisso e supera a MST: 1) reduzindo em 8% o número de métricas monitoradas, 2) aumentando em 10% o percentual de falhas detectadas e 3) reduzindo também em 10% a taxa de falso-positivos. Naturalmente, esses valores podem ser diferentes em outros cenários, mas os benefícios da abordagem PCTN-MST tendem a persistir.

5. Conclusão e Trabalhos Futuros

Neste trabalho, introduzimos a correlação parcial para selecionar correlações e métricas monitoradas que são empregadas na detecção de falhas em aplicações *Web* multi-camadas. A redução da quantidade de correlações e métricas, desde que oferecendo acurácia adequada, traz benefícios através da redução do consumo de recursos de centros de dados. Através de experimentos em um ambiente de testes real, mostramos que uma de nossas abordagens consegue superar uma estratégia estado-da-arte, gerando uma rede de correlações monitoradas menor e com acurácia superior. Em nossa abordagem, as correlações foram extraídas a partir de procedimentos *offline*. No entanto, em ambientes que apresentem carga de trabalho com alta dinamicidade, estratégias para detectar mudanças relevantes na carga de trabalho e a atualização dos modelos de forma *online* podem trazer benefícios e serão estudadas em trabalhos futuros.

Agradecimentos

Este trabalho foi parcialmente financiado por FAPEG, RNP e CNPq.

Referências

Aceto, G., Botta, A., De Donato, W., and Pescapè, A. (2013). Cloud monitoring: A survey. *Computer Networks*, 57(9):2093–2115.

Baba, K., Shibata, R., and Sibuya, M. (2004). Partial correlation and conditional correlation as measures of conditional independence. *Australian & New Zealand Journal of Statistics*, 46(4):657–664.

Bezenek, T., Cain, T., Dickson, R., Heil, T., Martin, M., McCurdy, C., Rajwar, R., Weglarz, E., Zilles, C., and Lipasti, M. (2014). Java TPC-W Implementation Distribution. `http://pharm.ece.wisc.edu/tpcw.shtml`. [Último acesso: 21-Set-2014].

Chen, H., Jiang, G., Yoshihira, K., and Saxena, A. (2010). Invariants based failure diagnosis in distributed computing systems. In *Reliable Distributed Systems, 2010 29th IEEE Symposium on*, pages 160–166.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 2nd edition edition.

De La Fuente, A., Bing, N., Hoeschele, I., and Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18):3565–3574.

Forster, F. (2014). collectd. `http://collectd.org`. [Último acesso: 05-Dez-2014].

IBM (2014). IBM SmartCloud Analytics. `http://www-03.ibm.com/software/products/en/ibm-smartcloud-analytics---predictive-insights`. [Último acesso: 06-Dez-2014].

Jiang, G., Chen, H., and Yoshihira, K. (2006). Modeling and tracking of transaction flow dynamics for fault detection in complex systems. *IEEE Trans. Dependable Secur. Comput.*, 3(4):312–326.

Jiang, M., Munawar, M. A., Reidemeister, T., and Ward, P. A. (2009). System monitoring with metric-correlation models: problems and solutions. In *Proceedings of the 6th international conference on Autonomic computing*, pages 13–22.

Kenett, D. Y., Tumminello, M., Madi, A., Gur-Gershgoren, G., Mantegna, R. N., and Ben-Jacob, E. (2010). Dominating clasp of the financial sector revealed by partial correlation analysis of the stock market. *PLoS ONE*, 5(12).

Magalhaes, J. and Silva, L. (2010). Detection of performance anomalies in web-based applications. In *Network Computing and Applications (NCA), 2010 9th IEEE International Symposium on*, pages 60–67.

Menasce, D. (2002). Tpc-w: a benchmark for e-commerce. *Internet Computing, IEEE*, 6(3):83–87.

Mi, N., Casale, G., Cherkasova, L., and Smirni, E. (2010). Sizing multi-tier systems with temporal dependence: benchmarks and analytic models. *Journal of Internet Services and Applications*, 1(2):117–134.

Microsoft (2013). System Center Operations Manager. `http://technet.microsoft.com/en-us/systemcenter/bb497976`. [Último acesso: 06-Dez-2014].

Munawar, M. A., Jiang, M., Reidemeister, T., and Ward, P. A. (2009). Filtering System Metrics for Minimal Correlation-Based Self-Monitoring. In *Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pages 233–242.

Peiris, M., Hill, J. H., Thelin, J., Bykov, S., Kliot, G., and Konig, C. (2014). Pad: Performance anomaly detection in multi-server distributed systems. In *7th IEEE International Conference on Cloud Computing (IEEE Cloud 2014)*.

Simic, B. (2014). Trac research. `http://www.new.trac-research.com`. [Último acesso: 10-Ago-2014].

Wang, C., Kavulya, S. P., Tan, J., Hu, L., Kutare, M., Kasick, M., Schwan, K., Narasimhan, P., and Gandhi, R. (2013). Performance troubleshooting in data centers: An annotated bibliography? *SIGOPS Oper. Syst. Rev.*, 47(3):50–62.

Wang, T., Wei, J., Zhang, W., Zhong, H., and Huang, T. (2014). Workload-aware anomaly detection for web applications. *The Journal of Systems and Software*, 89:19–32.

Zhang, Q., Cherkasova, L., and Smirni, E. (2007). A Regression-Based Analytic Model for Dynamic Resource Provisioning of Multi-Tier Applications. In *Proceedings of the Fourth International Conference on Autonomic Computing*, pages 27–36.