Impacto de Métricas de QoS no Desempenho de Detectores Auto-gerenciáveis de Defeitos para Sistemas Distribuídos

Alirio Santos de Sá[1], **Raimundo José de Araújo Macêdo**[1]

[1]Laboratório de Sistemas Distribuídos (LaSiD) / Departamento de Ciência da Computação
Universidade Federal da Bahia (UFBA) / Campus Ondina - Salvador - BA - Brasil

{aliriosa,macedo}@ufba.br

Abstract. *Self-manageable failure detectors have the ability to self-configure their operating parameters from QoS goals dynamically defined at runtime. Such detectors are intended for failure detection in modern distributed systems (such as grids and clouds), where the dynamicity of the computing environment and application requirements makes the configuration of the detection service a great challenge - which can compromise SLAs in terms of response time and availability. This paper shows for the first time, from experimental analyses, how distinct QoS metrics used to configure failure detector goals impact on the actual delivered QoS detection. Such analyses are important to allow , for instance, that dynamic changes in SLAs be properly planned to follow changes in the operational conditions and also in SLAs of upper layer services that make use of the failure detectors.*

Resumo. *Detectores de defeitos auto-gerenciáveis são capazes de auto-configurar dinamicamente seus parâmetros operacionais a partir de métricas de QoS de detecção definidas em tempo de execução pelas aplicações. Estes detectores são concebidos para o suporte à detecção de defeitos em sistemas distribuídos modernos (e.g. plataformas de grid e cloud computing), em que a dinamicidade dos ambientes de execução e dos requisitos das aplicações torna a configuração do serviço de detecção um grande desafio, podendo comprometer SLAs em termos do tempo de resposta ou da disponibilidade das aplicações, por exemplo. Neste artigo é mostrado pela primeira vez, a partir de análises experimentais, como as métricas de QoS usadas na configuração desses detectores auto-gerenciáveis impactam no desempenho exposto pelos mesmos. Estas análises são importantes para permitir, por exemplo, que mudanças dinâmicas nos SLAs de detecção sejam planejadas adequadamente para acompanhar mudanças operacionais do ambiente computacional e em SLAs de serviços de mais alto nível que utilizem os detectores.*

1. Introdução

Provedores de infra-estrutura de tecnologia da informação dos dias atuais (*data centers*, *grids*, etc.) utilizam diferentes componentes de hardware e software com intuito de prover serviços disponíveis, seguros e escaláveis para aplicações com necessidades variadas em termos de qualidade de serviço (QoS). Para serem bem sucedidas em seus propósitos, tais infra-estruturas precisam estar equipadas com mecanismos que garantam rápida reação e recuperação em contingências de falhas, maximizando a

disponibilidade dos serviços. Nesse contexto, detectores de defeitos têm papel fundamental uma vez que são utilizados para ativar processos de recuperação. Por exemplo, em um esquema de replicação passiva, o defeito de uma réplica primária precisa ser prontamente detectado para que uma das réplicas secundárias assuma o papel da réplica faltosa com o mínimo de impacto para as aplicações ou serviços distribuídos. Conseqüentemente, tais mecanismos vêm sendo objeto de intensa pesquisa nas últimas décadas [Chandra and Toueg 1996, Macêdo 2000, Chen et al. 2002, Bertier et al. 2002, Macêdo and Lima 2004, Nunes and Jansch-Pôrto 2004, Falai and Bondavalli 2005, Xiong et al. 2006, Satzger et al. 2008].

Na detecção de defeitos por *crash*, o foco principal das pesquisas, geralmente considera que processos monitorados enviam periodicamente mensagens, ditas *heartbeats*, as quais indicam seu estado para processos monitores. O monitor determina um intervalo de tempo (i.e. *timeout* de detecção) durante o qual esperará pela chegada do *heartbeat* correspondente. Se o *heartbeat* não chega dentro do *timeout* de detecção, o processo monitor suspeitará da falha do processo monitorado. Esse modelo de detecção de defeitos depende das restrições temporais relacionadas ao modelo do sistema. Em um modelo puramente assíncrono, os limites temporais para o processamento e para a transmissão das mensagens são desconhecidos [Fischer et al. 1985], o que torna impossível solucionar certos problemas de tolerância a falhas de forma determinística, como o consenso [Fischer et al. 1985], bem como a implementação de detectores de defeitos perfeitos. Para contornar essa impossibilidade, projetistas passaram a considerar modelos parcialmente síncronos em que o sistema se estabiliza em algum momento desconhecido [Dwork et al. 1988]. Para lidar com detecção de defeitos nesses modelos, Chandra e Toueg em [Chandra and Toueg 1996] introduziram o conceito de detectores de defeitos não confiáveis. Esses detectores são ditos não confiáveis, pois podem apontar como defeituosos, processos corretos, e, por outro lado, deixar de apontar processos que efetivamente falharam. [Chandra and Toueg 1996] demonstraram como, encapsulando certo nível de sincronia, detectores de defeitos não confiáveis podem ser usados para solucionar problemas fundamentais em sistemas distribuídos.

Apesar da importância do trabalho de [Chandra and Toueg 1996], a ausência de limites temporais dos modelos assíncronos impõe grandes desafios práticos para a implementação de detectores de defeitos. Um desses desafios é decidir valores apropriados para o *timeout* de detecção. *Timeouts* muito longos tornam a detecção lenta e compromete a resposta do sistema durante a ocorrência de falhas. Por outro lado, *timeouts* muito curtos podem degradar a confiabilidade do detector e prejudicar o desempenho do sistema – uma vez que muitos algoritmos e protocolos, que utilizam a informação do detector de defeitos, podem realizar processamento e troca de mensagens adicionais por conta de falsas suspeitas de falhas. Para melhorar a precisão de detecção, surgiram propostas de detectores adaptativos de defeitos, como em [Macêdo 2000], que propõe um mecanismo denominado CTI (*Connectivity Time Indicator*), o qual é inserido em uma abordagem de detecção de defeitos não confiável, com o intuito de sugerir *timeouts* de detecção dinâmicos – que variam de acordo com as condições de carga do ambiente distribuído. Com o mesmo propósito, outros trabalhos propõem o uso de estimadores de *timeouts* na implementação de detectores de defeitos – como por exemplo, [Bertier et al. 2002], [Macêdo and Lima 2004], [Nunes and Jansch-Pôrto 2004], [Falai and Bondavalli 2005], entre outros.

Com o propósito de avaliar o comportamento de detectores não confiáveis, [Chen et al. 2002] definiram métricas de QoS para detecção, as quais têm sido usadas desde então para avaliar a velocidade e a precisão de diferentes implementações de detectores de defeitos. Com isso, o trabalho do projetista é definir um período de monitoramento e usar um estimador de *timeout* que consiga entregar um serviço de detecção de defeitos com um nível de QoS adequado aos requisitos das aplicações. Entretanto, o ajuste do período de monitoramento tem recebido pouca atenção da literatura. Períodos de monitoramento muito curtos podem incrementar demasiadamente o consumo de recursos computacionais, comprometendo o tempo de resposta das aplicações e diminuindo a eficiência e a velocidade dos mecanismos de detecção de defeitos e de recuperação. Mais ainda, a escolha de períodos adequados se complicam em ambientes computacionais altamente dinâmicos, como *cloud computing*. Nesses ambientes é imperativo compatibilizar o custo do serviço de detecção com as características dinâmicas dos ambientes computacionais e de suas aplicações, podendo comprometer SLAs (*Service Level Agreement*) em termos do tempo de resposta ou da disponibilidade das aplicações. Somente muito recentemente apareceram propostas de detectores de defeitos capazes de auto-configurar seus parâmetros operacionais, inclusive o período de monitoramento, em tempo de execução, em resposta às mudanças no ambiente computacional ou nas aplicações, observando, para tanto, os requisitos de QoS definidos pelo usuário [de Sá and Macêdo 2010a, de Sá and Macêdo 2010b].

De outro lado, compatibilizar SLAs com a QoS entregue dinamicamente pelo detector não é um tarefa trivial, e às vezes mesmo impossível. Por exemplo, é impossível gerar detecções muito rápidas em ambientes computacionais trabalhando acima da carga rotineira (em sobrecarga), uma vez que detecção rápida implica em uma maior freqüência de *heartbeats*, que por sua vez, gera ainda mais carga na rede e nos servidores - piorando ainda mais a velocidade de detecção. Ou seja, às vezes SLAs podem evidenciar objetivos contraditórios e difíceis de serem conciliados. Por conta disso, surge o desafio de um estudo mais criterioso do impacto de SLAs no desempenho dos detectores, de modo a auxiliar os projetistas na escolha das combinações potencialmente mais bem sucedidas de parâmetros de QoS para a especificação de SLAs.

A partir de análises experimentais, este artigo responde a esse desafio, mostrando, pela primeira vez – até onde sabemos–, como as métricas de QoS usadas na configuração de detectores auto-gerenciáveis impactam no desempenho exposto pelos mesmos. Estas análises são importantes para permitir, por exemplo, que mudanças dinâmicas nos SLAs de detecção sejam planejadas adequadamente para acompanhar mudanças operacionais do ambiente computacional e em SLAs de serviços de mais alto nível que utilizem os detectores.

O resto deste artigo se organiza da seguinte forma. A Seção 2 descreve trabalhos relacionados. A Seção 3 apresenta o modelo do sistema. A Seção 4 faz uma rápida apresentação do detector de defeitos autonômico. A Seção 5 apresenta a avaliação de desempenho realizada e a discussão sobre o impacto das métricas de QoS de detecção. Finalmente, a Seção 6 tece as considerações finais.

2. Trabalhos Relacionados

Avaliação de detectores de defeitos baseada em métricas de QoS foram tratadas em diversos trabalhos. No entanto, a maioria das avaliações está focada nos *timeouts* de detecção, com pouquíssimas contribuições em relação a avaliação dos períodos de *heartbeats*, especialmente em ambientes altamente dinâmicos, cujas características de configuração e carga variam de forma não previsível.

Em [Chen et al. 2002] foi proposto um procedimento para a configuração *off-line* dos detectores de defeitos. Esses mesmos pesquisadores sugerem que tal procedimento de configuração pode ser re-executado durante o funcionamento do detector, quando as características de carga do ambiente computacional mudam. Todavia, os efeitos dessa re-execução no desempenho dos detectores de defeitos não foram avaliados.

Os autores em [Bertier et al. 2002] comentam brevemente um procedimento baseado em consenso para ajustar dinamicamente o período de monitoramento quando certas condições de carga são verificadas. Entretanto, esses autores não detalham a solução, nem avaliam a mesma considerando métricas de QoS de detecção.

[Mills et al. 2004], [Xiong et al. 2006] e [So and Sirer 2007] exploram a configuração dinâmica de detectores de defeitos. Entretanto, esses pesquisadores consideram que o comportamento do ambiente computacional não muda e não demonstram como dinamicamente configurar os detectores de defeitos usando métricas de QoS, como tempo de detecção, duração da falsa suspeita e intervalo entre falsas suspeitas.

Recentemente, [Dixit and Casimiro 2010] propuseram uma abordagem de detecção de defeitos que usa métricas de QoS para o ajuste do *timeout* de detecção. Entretanto, esses pesquisadores não consideram o ajuste dinâmico do período de monitoramento, outro aspecto importante para a adequação do custo computacional relacionado ao serviço de detecção de defeitos.

Em [de Sá and Macêdo 2010a] e [de Sá and Macêdo 2010b] propusemos detectores de defeitos capazes de ajustar dinamicamente seus parâmetros operacionais para atender a dinamicidade dos ambientes distribuídos modernos (e.g. *clouds* e *grids*). Estes detectores são os primeiros a dotar o serviço de detecção de defeitos da habilidade de configurar dinamicamente os seus parâmetros operacionais, considerando: métricas de QoS de detecção dinamicamente definidas pelo usuário; e o custo da detecção associado ao uso de recursos computacionais em ambientes distribuídos dinâmicos e abertos.

Todos esses trabalho elencados avaliam os respectivos detectores a partir de métricas de QoS de detecção. Contudo, até onde sabemos, nenhum trabalho existente na literatura avalia qual o impacto de divergentes SLAs, expressas também em QoS de detecção, sobre o desempenho dos detectores.

3. Modelo de Sistema

Um sistema distribuído é constituído por um conjunto finito de n processos $\Pi = \{p_1, p_2, ..., p_n\}$. Esses processos são interconectados através de canais de comunicação não confiáveis, os quais podem duplicar ou perder mensagens. Se uma mensagem enviada por um processo do sistema é corrompida, a mesma será descartada. Além disso, se um processo p_i envia, para um processo correto p_j, uma mesma mensagem sucessivas

vezes, em algum momento tal mensagem será recebida com sucesso por p_j – i.e. os processos interagem através de canais de comunicação do tipo *fair-lossy*. Não são assumidos limites temporais para a transmissão ou processamento das mensagens.

Os processos do sistema podem falhar por *crash* – falhas bizantinas [Lamport et al. 1982] não são consideradas. Cada processo tem acesso a um módulo local de um serviço de detecção de defeitos. Este serviço de detecção de defeitos utiliza o estilo de monitoramento *pull*, no qual um processo monitor p_i periodicamente verifica o estado de um processo monitorado p_j, enviando um mensagem de *"are you alive?"*(aya) e, então, p_j deve responder usando uma mensagem chamada *heartbeat* (hb) ou *"I'm alive!"*.

4. Visão Geral dos Detectores Autonômicos de Defeitos

A detecção autonômica de defeitos [de Sá and Macêdo 2010a, de Sá and Macêdo 2010b] considera a implementação de um gestor autonômico (ou controlador), o qual observa o comportamento do serviço básico de detecção de defeitos (elemento gerenciado ou planta). Então, baseado nos SLAs de detecção, esse gestor autonômico calcula o período de monitoramento (δ) e o *timeout* de detecção (rto), os quais são usados por um processo monitor p_i para checar o estado de um processo monitorado p_j (ver Figura 1).

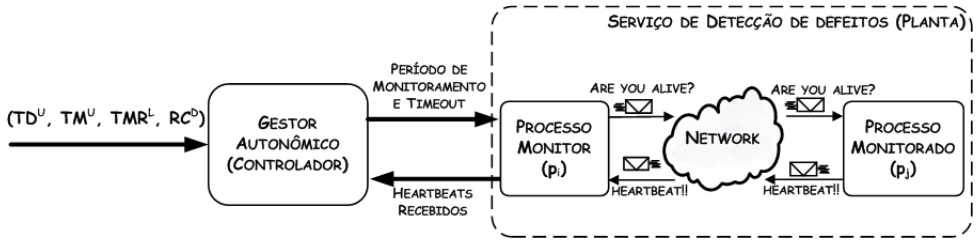

Figura 1. Abordagem de detecção autonômica de defeitos

O SLA de detecção é definido a partir de restrições relacionadas às métricas de QoS de detecção proposta por [Chen et al. 2002]: *Tempo de detecção* (TD) – intervalo de tempo necessário para que um processo monitor (p_i) suspeite a falha de um processo monitorado (p_j); *Duração da falsa suspeita de falha* (TM) – intervalo de tempo que o detector leva para corrigir uma falsa suspeita de falha; e *Intervalo entre falsas suspeitas de falhas* (TMR) – intervalo de tempo entre duas falsas suspeitas consecutivas. Assim, as restrições do SLA de detecção são definidas usando o tempo máximo de detecção (TD^U), a duração máxima da falsa suspeita (TM^U) e o intervalo mínimo entre falsas suspeitas (TMR^L). Em [de Sá and Macêdo 2010a] e [de Sá and Macêdo 2010b], TM e TMR são consideradas como métricas de confiabilidade de detecção e utilizadas para estimar a qualidade de serviço em termos da disponibilidade de detecção (AV), usando: $AV = (TMR - TM)/TMR$. Então, os autores também definem o requisito de disponibilidade mínima de detecção (AV^L) como: $AV^L = (TMR^L - TM^U)/TMR^L$.

Além destas restrições de QoS de detecção (i.e. TD^U, TM^U e TMR^L), [de Sá and Macêdo 2010a] e [de Sá and Macêdo 2010b] também incorporam ao SLA uma restrição relacionada ao uso de recursos computacionais, dita RC^D (percentual de consumo de recursos).

O gestor autonômico é composto por três tarefas básicas: (i) **sensoriamento** de características do ambiente computacional e da QoS do serviço de detecção de defeitos –

para observar o atendimento dos SLAs definidos pelos usuários; (ii) **regulação do *timeout* de detecção** – para atender o compromisso entre a precisão (TM e TMR) e a velocidade (TD) das detecções; (iii) **regulação do período de monitoramento** – para atender o compromisso entre a velocidade (TD) e o custo computacional (RC) das detecções.

Ambas as propostas de detecção autonômicas apresentadas em [de Sá and Macêdo 2010a] e [de Sá and Macêdo 2010b] utilizam as mesmas estratégias de sensoriamento e de regulação de *timeout* de detecção. Todavia, estas propostas se diferenciam pela estratégia usada na regulação do período de monitoramento. [de Sá and Macêdo 2010a] usa uma lei de controle P, na qual o ajuste no período é proporcional ao desvio entre os percentuais de consumo de recursos atual (RC) e desejado (RC^D). [de Sá and Macêdo 2010b] usa uma lei de controle PI, na qual o ajuste no período considera uma componente proporcional ao desvio entre RC e RC^D e outra componente que é proporcional a integral de tais desvios. As abordagens de [de Sá and Macêdo 2010a] e [de Sá and Macêdo 2010b] são denominadas RBS e RBL, respectivamente.

5. Avaliação do Impacto de SLAs de Detecção no Desempenho dos Detectores

Nesta seção avalia-se o impacto da variação de cada parâmetro de SLA de detecção no desempenho dos detectores autonômicos. Antes da análise de impacto, os experimentos são detalhados na sub-seções que seguem.

5.1. Descrição do Ambiente de Simulação e Experimentos

Os experimentos foram realizados por simulação com o auxílio do pacote *Matlab / Simulink / TrueTime 1.5*[Henriksson and Cervin 2003]. O ambiente simulado considera três computadores, ditos c_1, c_2 e c_3, conectados através de uma rede *Switched Ethernet* com taxa nominal de transferência de $10Mbps$, *buffer* de $1MB$ e mecanismo de descarte de mensagens em caso de *buffer overflow*. As mensages trocadas entre os processos do ambiente têm tamanho fixo e igual a $1536bits$ – equivalente a três vezes o tamanho mínimo de um quadro *Ethernet*. Tais parâmetros de rede são escolhidos de modo a se obter tempos de simulações adequados em experimentos com condições extremas de carga (i.e. baixas ou altas condições de uso de rede).

O processo em c_1 monitora defeitos do processo em c_2. Em c_3, um processo com ativação periódica, denominado p_3, é utilizado para gerar rajadas aleatórias de tráfego na rede. Essas rajadas são geradas usando uma distribuição de Bernoulli para decidir em que ativações do processo p_3 as rajadas são geradas. O processo p_3 é ativado a cada $0,1536ms$, i.e. 1536 bits dividido por $10Mbps$ – o que equivale ao tempo necessário para transmitir um quadro de 1536 bits. A probabilidade de envio de rajadas determina o percentual de ocupação de banda de rede e é controlada pela variável $BW \in [0,1]$ – por exemplo: se o valor de $BW = 0$, então o processo p_3 não emite rajadas (ocupação de rede zero); por outro lado, se $BW = 1$, então as rajadas de p_3 ocupam 100% da banda de rede. A variável BW é variada durante a execução dos experimentos, de acordo com a ocupação desejada.

Para garantir justiça em relação ao número de mensagens trocadas durante a avaliação dos detectores, cada experimento pode ter tempos de simulação diferentes, mas sempre executam até que sejam transferidas 10^4 mensagens de monitoramento.

5.2. Fatores, Métricas de Desempenho e Detectores Autonômicos Avaliados

Os fatores são os parâmetros de interesse que podem afetar o desempenho e são variados durante os experimentos [Jain 1991]. Durante a avaliação experimental são considerados os seguintes fatores de desempenho: [a] Os parâmetros de configuração dos detectores autonômicos de defeitos – i.e. os parâmetros que compõem o SLA de detecção (ou seja, TD^U, AV^L e RC^D, ver Seção 4); e [b] as condições de carga impostas pelas aplicações dos usuário no ambiente distribuído – representadas nos experimentos pelos valores definidos para a variável BW (ver Subseção 5.1).

As métricas definem os critérios usados para quantificação do desempenho dos elementos computacionais avaliados [Jain 1991] – e.g. os detectores autonômicos. Assim, como métricas de desempenho são usadas: [a] as métricas de qualidade de serviço de detecção de defeitos – i.e. TD, TM e TMR; e [b] a taxa de falsas suspeitas cometidas por cada detector autonômico (RM).

A taxa de falsas suspeitas é definida como sendo a relação entre o total de falsas suspeitas cometidas (n_f) por um detector de defeitos e o respectivo total de verificações de estado (n_v) realizadas pelo mesmo, isto é: $RM = n_f/n_v$. Esta métrica é usada como uma alternativa a medição simples do número de falsas suspeitas, como é tradicionalmente realizado nas avaliações de desempenho da literatura. Isto porque, a taxa de falsa suspeita é uma métrica mais justa quando se considera detectores de defeitos com períodos de monitoramento variados. Como exemplo, considere uma estratégia de detecção de defeitos configurada com *timeout* de detecção fixado. Se esta estratégia usa períodos de monitoramento mais curtos naturalmente tende a realizar mais verificações dos estados dos processos que com períodos mais longos – podendo cometer mais falsas suspeitas. Note que, neste caso, um maior ou menor período significa apenas que o algoritmo usado pela estratégia é executado um número maior ou menor de vezes, não significando necessariamente que a mesma desempenhe melhor ou pior o seu papel.

Os experimentos consideram as duas abordagens de detecção autonômica propostas (i.e. RBL e RBS, ver Seção 4). Estas abordagens encapsulam diferentes detectores adaptativos, compondo diferentes versões de detectores autonômicos – ver Tabela 1.

Tabela 1. Diferentes versões dos detectores autonômicos considerados

Versão do AFD	Descrição
$JAFD$-RBL	Detector autonômico baseado na abordagem de regulação de período RBL, encapsulando o estimador de *timeouts* de [Jacobson 1988]
$BAFD$-RBL	Detector autonômico baseado na abordagem de regulação de período RBL, encapsulando o estimador de *timeouts* de [Bertier et al. 2002]
$JAFD$-RBS	Detector autonômico baseado na abordagem de regulação de período RBS, encapsulando o estimador de *timeouts* de [Jacobson 1988]
$BAFD$-RBS	Detector autonômico baseado na abordagem de regulação de período RBS, encapsulando o estimador de *timeouts* de [Bertier et al. 2002]

5.3. Metodologia de Avaliação Experimental Adotada

Para análise experimental do impacto dos parâmetros de SLA no desempenho dos detectores autonômicos, utiliza-se a metodologia de avaliação experimental 2^q_r fatorial – em que q e r referem-se, respectivamente, ao número de fatores e ao número de repetições dos experimentos, ver [Jain 1991]. Esta metodologia não propriamente identifica a relação

entre os parâmetros de configuração e o desempenho do detector autonômico, mas provê uma expectativa sobre o quão importante é cada um dos parâmetros de configuração para o desempenho desejado do sistema avaliado (no caso, os detectores autonômicos).

Nas análises são considerados os quatro fatores apresentados (i.e. $q = 4$), ou seja: RC^D, TD^U, AV^L e BW. Cada série de experimentos é repetida três vezes (i.e. $r = 3$). Portanto, foram realizados 2^4 experimentos, re-executados três vezes e considerando 10^4 amostras cada um – o que totaliza 480000 amostras (i.e. mensagens de monitoramento).

Na metodologia de análise experimental 2^q_r, os fatores considerados secundários[1] são fixados, enquanto que os fatores básicos (i.e. RC^D, TD^U, AV^L e BW) variam. Para tanto, estes fatores básicos são variados considerando dois valores extremos para cada um. Um destes valores determina impactos positivos em uma ou mais métricas de desempenho, enquanto o outro implica em impactos negativos (ver Tabela 2).

Tabela 2. Fatores e seus níveis de impactos

Apelido do fator	Nome do fator	Níveis de impacto	
		-	+
		Valores considerados	
x_1	TD^U	$0,3072ms$	$61,44ms$
x_2	AV^L	$0,0$	$1,0$
x_3	RC^D	$0,0$	$1,0$
x_4	BW	$0,9$	$0,0$

Por examplo, $0,3072ms$ é o atraso de ida-e-volta mínimo para cada mensagem de monitoramento. Se $TD^U = 0,3072ms$, então o gestor autonômico do detector não ajusta de forma apropriada o período de monitoramento sob altas condições de carga (i.e. $BW = 0,9$). Assim, espera-se que o detector autonômico possua um baixo desempenho em termos do tempo de detecção – com isso, associa-se um nível de impacto negativo para esse valor de TD^U. Por outro lado, para $TD^U = 61,44ms$, o qual representa 200 vezes o atraso de ida-e-volta mínimo, é esperado que o detector autonômico ajuste de forma adequada o período de monitoramento sob altas condições de carga, apresentando assim um bom desempenho em termos do tempo de detecção – neste caso, associa-se um nível de impacto positivo para esse valor de TD^U.

Considerações similares são realizadas para os demais fatores básicos apresentados na Tabela 2. Caso os níveis de impacto (i.e. nível positivo ou negativo de impacto) para cada um dos valores sejam diferentes daqueles supostos durante o projeto dos experimentos, os valores percentuais médios de impactos observados após a execução das análises terão valores negativos.

De acordo com a metodologia 2^q_r, na execução da avaliação experimental, todos os detectores são configurados considerando todas as combinações de valores dos fatores básicos definidos na Tabela 2. Então, os valores médios do desempenho em termos das métricas consideradas são observados. Em seguida, o percentual de variação em cada uma das métricas de desempenho são obtidos usando o método dos mínimos quadrados. Esses percentuais de variação compõem uma tabela que determina o percentual de impacto de cada um dos fatores básicos (e de cada uma de suas interações) no desempe-

[1] e.g. tipo de rede, taxa nominal de transferência da rede, tamanho do buffer de mensagens etc.

nho do detector autonômico considerado. Além disso, erros experimentais são calculados (a partir das repetições) e registrados. Esses erros experimentais dizem respeito aos impactos de fatores secundários que não foram considerados (ou variados) na avaliação experimental [Jain 1991]. A depender da magnitude dos erros experimentais novos experimentos podem ser conduzidos incorporando outros fatores que foram desconsiderados em uma avaliação preliminar. Para uma discussão mais detalhada sobre a metodologia de avaliação experimental 2^q_r fatorial, veja [Jain 1991].

5.4. Resultado da Avaliação Experimental

As tabelas 3, 4, 5 e 6 apresentam o percentual de impacto de cada um dos fatores básicos e de suas interações sobre o desempenho dos detectores autonômicos em termos do tempo de detecção (TD), da duração da falsa suspeita (TM), da taxa de falsas suspeitas (RM) e do intervalo entre falsas suspeitas (TMR), respectivamente. Nas tabelas, os valores das linhas somam 100% para cada coluna. A última linha de cada tabela apresenta os erros experimentais encontrados. As linhas em cinza representam os fatores e/ou interações com percentual de impacto relevante para o desempenho em termos da métrica considerada. Por fim, nestas tabelas, as interações entre fatores são representadas como (x, y), significando que os percentuais de impacto no desempenho, apresentados na respectiva linha, referem-se à combinação do fator x com o fator y.

5.4.1. Impacto dos Fatores e de suas Interações no Tempo de Detecção (TD)

A Tabela 3 apresenta o percentual de impacto dos fatores e de suas interações no desempenho em termos de TD para as diferentes versões do detector autonômico.

Tabela 3. Percentual de impacto no tempo de detecção

Fatores e Interações	$JAFD$-RBL	$BAFD$-RBL	$JAFD$-RBS	$BAFD$-RBS
AV^L	2	2	1	5
RC^D	33	35	14	23
TD^U	27	24	24	34
BW	1	1	13	4
(AV^L, RC^D)	0	0	7	1
(AV^L, TD^U)	1	1	2	0
(AV^L, BW)	0	0	10	2
(RC^D, TD^U)	35	36	13	24
(RC^D, BW)	0	0	0	1
(TD^U, BW)	0	0	4	0
(AV^L, RC^D, TD^U)	0	0	7	1
(AV^L, RC^D, BW)	0	0	0	1
(AV^L, TD^U, BW)	0	0	5	0
(RC^D, TD^U, BW)	0	0	0	1
(AV^L, RC^D, TD^U, BW)	0	0	0	1
Erro experimental	0	0	1	1

Nessa tabela, RC^D e TD^U e sua interação (RC^D, TD^U) possuem impacto relevante no desempenho de todas as versões. Estes fatores e suas interações somados impactam entre 53% e 95% do desempenho em termos do TD para cada versão do detector autonômico, o que indica que é possível explorar os parâmetros RC^D e TD^U para obter boas alternativas de desempenho em termos do tempo de detecção. Para os detectores

autonômicos na abordagem RBL, usando os estimadores de [Jacobson 1988] ($JAFD$-RBL) e de [Bertier et al. 2002] ($BAFD$-RBL), os impactos de RC^D e TD^U e de suas interações somam 95% do desempenho em termos do TD (em ambas as versões deste detector autonômico). Para os detectores autonômicos com a abordagem RBS, por outro lado, os fatores RC^D e TD^U e sua interação têm percentual de impacto de 53% e 81% para $JAFD$-RBS e $BAFD$-RBS, respectivamente.

As duas versões da abordagem RBS sofrem menos influência de RC^D e TD^U, por questões distintas. No caso de $JAFD$-RBS, existe uma forte influência dos fatores BW e AV^L e de suas interações. A soma dos impactos destes fatores e de suas interações impactam 48% no desempenho de $JAFD$-RBS em termos de TD. Isto porque, o estimador de [Jacobson 1988] se ajusta rapidamente à variação da carga, o que deixa o detector mais susceptível a falsas suspeitas que o estimador de [Bertier et al. 2002], por exemplo. Portanto, na versão $JAFD$-RBS, o regulador de *timeout* da abordagem autonômica atua mais – o que explica a influência de AV^L no tempo de detecção. Além disso, ação integral do controlador PI torna a regulação de período mais lenta nas versões de RBS. Consequentente, nos cenários em que a carga da aplicação muda de forma mais brusca (BW maiores), isto é somado com a carga introduzida pelo detector, ocasionando uma maior variabilidade dos atrasos, degrando o desempenho das versões dos detectores autonômicos RBS. Por conta disto, BW e suas interações com os demais fatores possuem um impacto mais significativo no desempenho destas versões do detector autonômico que no caso das versões com a abordagem RBL.

5.4.2. Impacto dos Fatores e de suas Interações na Duração da Falsa Suspeita (TM)

A Tabela 4 apresenta o percentual de impacto dos fatores e de suas interações no desempenho em termos de TM para as diferentes versões do detector autonômico.

Tabela 4. Percentual de impacto na duração da falsa suspeita

Fatores e Interações	$JAFD$-RBL	$BAFD$-RBL	$JAFD$-RBS	$BAFD$-RBS
AV^L	24	26	29	25
RC^D	0	2	3	5
TD^U	34	34	19	29
BW	1	0	0	2
(AV^L, RC^D)	1	0	0	2
(AV^L, TD^U)	22	20	16	22
(AV^L, BW)	0	3	6	0
(RC^D, TD^U)	3	0	1	0
(RC^D, BW)	0	2	6	5
(TD^U, BW)	1	3	7	0
(AV^L, RC^D, TD^U)	1	0	0	1
(AV^L, RC^D, BW)	1	0	1	2
(AV^L, TD^U, BW)	0	5	6	0
(RC^D, TD^U, BW)	3	0	3	0
(AV^L, RC^D, TD^U, BW)	1	0	0	1
Erro experimental	7	6	4	4

Nessa tabela, AV^L e TD^U e sua interação (AV^L, TD^U) possuem impactos relevantes no desempenho. Estes fatores e sua interação somados representam entre 64% e 80% do desempenho em termos de TM para cada versão do detector autonômico. Isto

indica que é possível explorar os parâmetros AV^L e TD^U para obter boas alternativas de desempenho em termos da duração da falsa suspeita. Para os detectores autonômicos na abordagem RBL, usando os estimadores de [Jacobson 1988] ($JAFD\text{-}RBL$) e de [Bertier et al. 2002] ($BAFD\text{-}RBL$), os impactos de AV^L e TD^U e de sua interação somam 80% do desempenho em termos de TM. Para os detectores autonômicos com a abordagem RBS, por outro lado, os fatores AV^L e TD^U e sua interação têm percentual de impacto de 64% e 76% para $JAFD\text{-}RBS$ e $BAFD\text{-}RBS$, respectivamente.

Dentre as versões do detector autonômico, AV^L possui maior impacto percentual na duração da falsa suspeita quando se observa $JAFD\text{-}RBS$, o que reforça a explanação apresentada na Seção 5.4.1 sobre a influência de tal métrica em TD. Mais precisamente, Quando o gestor autonômico atua no *timeout*, isto reduz o número de falsas suspeitas, mas impacta em TD e implica em TM maiores, em alguns casos.

5.4.3. Impacto dos Fatores e de suas Interações na Taxa de Falsas Suspeitas (RM)

A Tabela 5 apresenta o percentual de impacto dos fatores e de suas interações no desempenho em termos de RM para as diferentes versões do detector autonômico.

Tabela 5. Percentual de impacto na taxa de falsas suspeitas

Fatores e Interações	$JAFD\text{-}RBL$	$BAFD\text{-}RBL$	$JAFD\text{-}RBS$	$BAFD\text{-}RBS$
AV^L	44	52	48	46
RC^D	3	2	2	4
TD^U	4	8	4	4
BW	6	3	6	4
(AV^L, RC^D)	2	2	2	2
(AV^L, TD^U)	6	9	8	7
(AV^L, BW)	5	3	6	3
(RC^D, TD^U)	3	2	3	4
(RC^D, BW)	2	1	2	2
(TD^U, BW)	2	2	2	1
(AV^L, RC^D, TD^U)	2	2	3	2
(AV^L, RC^D, BW)	1	1	1	1
(AV^L, TD^U, BW)	2	1	2	1
(RC^D, TD^U, BW)	2	1	3	2
(AV^L, RC^D, TD^U, BW)	1	1	2	1
Erro experimental	16	9	7	17

Nesta tabela, AV^L é um fator dominante do desempenho de todas as versões dos detectores autonômicos. Este fator apresenta percentuais de impacto em RM de 44%, 52%, 48% e 46% para $JAFD\text{-}RBL$, $BAFD\text{-}RBL$, $JAFD\text{-}RBS$ e $BAFD\text{-}RBS$, respectivamente. Além disso, no caso da métrica RM, os percentuais de impacto de RC^D, de TD^U e de BW e de suas interações somados significam algo entre 37% e 45%. Isto significa que uma combinação apropriada dos fatores pode levar a um melhor desempenho dos detectores autonômicos em termos de RM.

Note que o erro experimental observado está entre 9% e 17%, o que é um pouco mais representativo que os observados para TD e TM – significando que os fatores secundários passam a ser um pouco mais relevantes quando o desempenho em termos de RM é considerado.

5.4.4. Impacto dos Fatores e de suas Interações no Intervalo entre Falsas Suspeitas

A Tabela 6 apresenta o percentual de impacto dos fatores e de suas interações no desempenho em termos de TMR para as diferentes versões do detector autonômico.

Tabela 6. Percentual de impacto no intervalo entre falsas suspeitas

Fatores e Interações	$JAFD\text{-}RBL$	$BAFD\text{-}RBL$	$JAFD\text{-}RBS$	$BAFD\text{-}RBS$
AV^L	5	3	2	3
RC^D	13	13	11	13
TD^U	11	12	10	12
BW	5	8	10	9
(AV^L, RC^D)	4	2	1	2
(AV^L, TD^U)	3	2	1	1
(AV^L, BW)	1	1	2	1
(RC^D, TD^U)	12	13	11	14
(RC^D, BW)	5	8	10	8
(TD^U, BW)	4	8	10	8
(AV^L, RC^D, TD^U)	4	2	1	2
(AV^L, RC^D, BW)	1	1	2	1
(AV^L, TD^U, BW)	1	1	2	1
(RC^D, TD^U, BW)	4	7	10	9
(AV^L, RC^D, TD^U, BW)	1	1	2	1
Erro experimental	24	20	18	17

Nestes experimentos, nenhum dos fatores ligados à configuração do detector autonômico responde sozinho pelo desempenho percentual em termos de TMR – todos esses fatores sozinhos possuem impacto percentual abaixo de 15%. Isto se deve ao fato de o desempenho em termos de TMR dos detectores depender não apenas dos fatores básicos relacionados a configuração (i.e. TD^U, RC^D e AV^L), mas também do modo como esses fatores se relacionam com as diferentes condições de carga (representado pelo fator BW).

Todavia, os efeitos de TD^U e RC^D e de sua interação (RC^D, TD^U) são significativos em termos dessa métrica – apresentando somatórios de impactos percentuais entre 32% e 39%. Isto porque, estas métricas estão diretamente ligadas à regulação do período de monitoramento, um fator determinante para o desempenho em termos de TMR: quanto maior o período de monitoramento maior o TMR (independente dos demais fatores). Quando, além de RC^D, TD^U e (RC^D, TD^U), AV^L e suas interações são somados, a faixa percentual de impacto se acomoda entre 35% e 47% – isto é, existe um aumento no impacto de 3 a 15%, a depender da versão do detector autonômico.

Por outro lado, BW e suas interações com os demais fatores respondem entre 22% e 48% dos impactos percentuais no desempenho das diferentes versões do detector autonômico. Isto porque, dada uma variação na carga, o gestor autonômico deve: (a) encontrar um período de monitoramento que reduza a interferência do detector nas variações de carga; e (b) realizar correções no *timeout* de detecção até que o detector possua a confiabilidade desejada. Esse intervalo transiente no qual o detector encontra a sintonia correta para o período de monitoramento, implica em maiores oscilações de carga, o que justifica a influência de BW e de suas interações no desempenho do detector autonômico – observe que BW responde sozinho com impactos percentuais entre 5% e 10% do desempenho, o que é inferior aos impactos percentuais de RC^D e TD^D sozinhos, mas é superior ao impacto de AV^L.

Por fim, note que o erro experimental representa entre 17% e 24%, significando que existem fatores secundários que possuem impacto percentual significativo em termos de TMR – por exemplo, os parâmetros de configuração dos estimadores de *timeout* de [Jacobson 1988] e de [Bertier et al. 2002].

6. Considerações Finais

Este artigo avaliou, através de análises experimentais, o impacto de SLAs no desempenho em termos de QoS de detectores de defeitos, sendo o primeiro estudo sobre o assunto disponível na literatura – até onde sabemos.

A partir dos experimentos realizados conclui-se que as restrições descritas no SLA de detecção se relacionam fortemente com a QoS de detecção entregue pelos detectores autonômicos. Neste sentido, TD^U e RC^D possuem um alto percentual de impacto no desempenho em termos do tempo de detecção. Além disso, AV^L e TD^U impactam também de forma bastante significativa no desempenho dos detectores em termos da duração da falsas suspeita. O desempenho em termos da taxa de falsas suspeitas, por sua vez, é completamente definida a partir de AV^L. Em termos dos intervalos entre falsas suspeitas, por outro lado, os impactos são divididos entre TD^U, AV^L, RC^D. Estas avaliações colaboram com os resultados obtidos em [de Sá and Macêdo 2010a] e [de Sá and Macêdo 2010b], servindo como mais uma validação destas propostas.

Uma vez que se tem uma expectativa do impacto das restrições do SLA sobre o desempenho em termos de QoS do detector, os resultados apresentados nos experimentos permitem que os projetistas possam desenvolver estratégias que definam de forma mais apropriada os valores das restrições dos SLAs de detecção.

Como trabalhos futuros considera-se o desenvolvimento de uma estratégia que permita mapear de forma adequada restrições de QoS no nível das aplicações (e.g. taxa de transações por segundo, tempo de resposta etc.) em restrições de QoS de detecção de defeitos. Os resultados destas análises serão usados na definição de classes de serviços para ambientes de computação em nuvens – em desenvolvimento no Laboratório de Sistemas Distribuídos (LaSiD/UFBA) no contexto do projeto JiTClouds/CTIC/RNP.

Referências

Bertier, M., Marin, O., and Sens, P. (2002). Implementation and performance evaluation of an adaptable failure detector. In *Proceedings of the International Conference on Dependable Systems and Networks*, pages 354–363, Washington, DC, USA. IEEE Computer Society.

Chandra, T. D. and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267.

Chen, W., Toueg, S., and Aguilera, M. K. (2002). On the quality of service of failure detectores. *IEEE Transactions on Computers*, 51(2):561–580.

de Sá, A. S. and Macêdo, R. J. A. (2010a). Detectores de defeitos autonômicos para sistemas distribuídos. In *Anais do XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, SBRC'2010, pages 785–798, Gramado, Brasil. SBRC / SBC.

de Sá, A. S. and Macêdo, R. J. A. (2010b). QoS self-configuring failure detectors for distributed systems. In *Proceedings of the 10th IFIP WG 6.1 International Conference on Distributed Ap-*

plications and Interoperable Systems (DAIS' 2010), volume 6115 of *Lecture Notes in Computer Science (LNCS)*, pages 126–140, Berlin, Heidelberg. Springer-Verlag.

Dixit, M. and Casimiro, A. (2010). Adaptare-fd: A dependability-oriented adaptive failure detector. In *Proc. of the 29th IEEE Symposium on Reliable Distributed Systems*, RDS 2010.

Dwork, C., Lynch, N., and Stockmeyer, L. (1988). Consensus in the presence of partial synchrony. *Journal of ACM*, 35:288–323.

Falai, L. and Bondavalli, A. (2005). Experimental evaluation of the qos failure detectors on wide area network. In *Proceedings of the 2005 International Conference On Dependable Systems And Networks (DSN'2005)*, pages 624–633, Yokohama, Japan. IEEE Computer Society.

Fischer, M. J., Lynch, N. A., and Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *Journal of ACM*, 32:374–382.

Henriksson, D. and Cervin, A. (2003). Truetime 1.5 - reference manual. Technical report, Department Of Automatic Control, Lund Institute Of Technology, Lund University, Sweden.

Jacobson, V. (1988). Congestion avoidance and control. In *Proc. of the Symp. on Communications Architectures and Protocols*, SIGCOMM '1988, pages 314–329, New York, NY, USA. ACM.

Jain, R. (1991). *The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling*. Addison-Wesley, USA.

Lamport, L., Shostak, R., and Pease, M. (1982). The byzantine generals problem. *ACM Transaction on Programming Languages Systems*, 4:382–401.

Macêdo, R. J. A. (2000). Failure detection in asynchronous distributed systems. In *Anais do II Workshop em Testes e Tolerância a Falhas*, pages 76–81, Curitiba, PR, Brazil. SBC.

Macêdo, R. J. A. and Lima, F. (2004). Improving the quality of service of failure detectors with SNMP and artificial neural networks. In *Anais do 22o Simpósio Brasileiro de Redes de Computadores (SBRC'2004)*, pages 583–586, Gramado, RS, Brazil. SBRC / SBC.

Mills, K., Rose, S., Quirolgico, S., Britton, M., and Tan, C. (2004). An autonomic failure-detection algorithm. In *Proceedings of the 4th International Workshop on Software and Performance*, WOSP'2004, pages 79–83, New York, NY, USA. ACM.

Nunes, R. C. and Jansch-Pôrto, I. (2004). Qos of timeout-based self-tuned failure detectors: The effects of the communication delay predictor and the safety margin. In *Proc. of the 2004 Int. Conference On Dependable Systems And Networks (DSN'2004)*, pages 753–761. IEEE.

Satzger, B., Pietzowski, A., Trumler, W., and Ungerer, T. (2008). A lazy monitoring approach for heartbeat-style failure detectors. In *Proc. of the 2008 Third International Conference on Availability, Reliability and Security*, pages 404–409, Washington, DC, USA. IEEE CS.

So, K. C. W. and Sirer, E. G. (2007). Latency and bandwidth-minimizing failure detectors. In *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, EuroSys'2007, pages 89–99, New York, NY, USA. ACM.

Xiong, N., Yang, Y., Chen, J., and He, Y. (2006). On the quality of service of failure detectors based on control theory. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, AINA'2006, pages 75–80. IEEE Computer Society.