

# Teste de desempenho em soluções SAP: um estudo de caso

Raul Nogueira Neves<sup>1</sup>, Maicon Bernardino<sup>1</sup>

<sup>1</sup>Laboratory of Empirical Studies in Software Engineering (LESSE)  
Software Engineer – Federal University of Pampa (Unipampa)  
97.546-550 – Alegrete – RS – Brazil

raulneves.aluno@unipampa.edu.br, bernardino@acm.org

**Abstract.** *This study presents the importance of software engineering in the development and updating of large-scale applications, using the SAP tool as a case study. SAP SE is the German multinational that has 99 of the 100 largest companies within its customer portfolio, as well as a multinational mining company. This study shows the scalability issues of one of the largest logistics operators in the world and the measures that have been taken to improve performance of the world's leading integrated management system.*

**Resumo.** *Este artigo apresenta a importância da engenharia de software no desenvolvimento e atualização de aplicações em larga escala, utilizando a ferramenta SAP como material de estudo. A SAP SE é a multinacional alemã que tem 99 das 100 maiores empresas dentro da sua carteira de clientes, dentre esses clientes está uma mineradora multinacional. O estudo analisa os problemas de escalabilidade de uma das maiores operadoras de logística do mundo. Os resultados apresentados serão as medidas a ser tomadas para melhorar o desempenho do principal sistema de gestão integrado do mundo.*

## 1. Introdução

A globalização e as necessidades de negócio hoje apresentadas fizeram com que grandes corporações aumentem, horizontalmente e verticalmente, no mercado que operam, gerando e consumindo dados em larga escala 24 horas por dia, 7 dias por semana. Hoje essas empresas se preocupam se todas as APIs funcionam sincronizadas tal qual uma sinfonia, enviando, recebendo e alterando informações do *core* a qual pertencem, sejam preços, fretes, estoques e outras tantas operações que acontecem por debaixo dos panos.

Assim como uma sinfonia, a preocupação de tais empresas é o tempo que essas requisições levam para se comunicar entre si e se as cadeias de informações que foram consultadas ou alteradas persistem em seus bancos de dados. Essa necessidade gerou uma carência de engenheiros de qualidade de software capazes não apenas de criar ecossistemas completos, com aplicações responsivas que interagem em tempo real com APIs de mercado e outras aplicações, mas também profissionais capazes de operar ambientes já criados e consolidados independente do ramo de negócios da empresa que possui a aplicação.

Por muitas vezes essas ferramentas são padrões de mercado, as quais tem o objetivo de melhorar um ou mais processos internos, garantindo que as regras de negócio do cliente funcionam e estão de acordo com a legislação vigente no país em que esta sendo

utilizado, o que se torna um desafio em softwares e empresas de larga escala que precisam de aplicações em diferentes partes do mundo.

Este artigo apresenta a primeira etapa da atualização de um ecossistema já existente e expõe a necessidade de uma multinacional do mercado de mineração que utiliza o maior fornecedor mundial de software para negócios, a amplamente conhecida SAP, mais especificamente sobre as transações IW21 e IW22 que pertencem ao módulo de planejamento de manutenção, cujo principal objetivo é tornar as operações dentro dos módulos inteligentes, confiáveis e sustentáveis.

O desafio proposto para a atualização do ambiente surgiu por meio da necessidade interna de agilidade na comunicação entre *APIs* em três localidades diferentes: Vitória no Brasil, Quebec no Canadá e Shulangu na China. Tais localidades estão localizadas completamente diferentes com estruturas diferentes, em caso de sobrecarga ou falha na operação, podem gerar prejuízos de bilhões de dólares. O estudo apresenta mais detalhes sobre a estrutura e suas peculiaridades, além de detalhes sobre os ciclos de testes de desempenho, tempo de execução e a comparação entre os ciclos e com base nas informações levantadas é possível refletir, sobre as possíveis soluções produtivas para a empresa solicitante.

## 2. Fundamentação Teórica

A garantia da qualidade (*Quality Assurance - QA*) é um processo abrangente com o objetivo de evitar falhas de qualidade, os Engenheiros de Software estão presentes em todos os estágios desde a arquitetura [Sommerville 2019], desenvolvimento, passando pelos testes até a entrega do produto ou serviço de software, o chamado ciclo de vida da aplicação. O objetivo de garantir que o produto ou serviço que está em desenvolvimento atenda os padrões de qualidade do ecossistema ao qual a aplicação pertence ou irá pertencer. Dentro desse abrangente processo existe uma categoria que exige pró-atividade para alcançar os diversos resultados possíveis, os testes de desempenho. Esses testes não estão procurando erros, falhas ou verificando se as regras de negócio foram ou não atendidas. O teste de desempenho tem a natureza preventiva focada em *user experience* e na melhoria contínua de processos e aplicações em geral.

### 2.1. Teste de Desempenho

Testes de desempenho estão geralmente relacionados a 5 pilares: medir, analisar, melhorar e repetir. Cada um desses pilares estão criteriosamente relacionados a velocidade, responsividade e estabilidade dos requisitos que foram propostos durante a criação do plano de testes. Os requisitos são diferentes quando aplicados em um teste funcional de uma aplicação não tem respostas tão simples quanto “passou ou não passou”, mas engloba análise completa de todas as variáveis de ambiente. Segundo [Weiss et al. 2013] é de extrema importância a identificação de problemas de desempenho em aplicações, antes que seus clientes percebam, pois a qualidade e o sucesso da sua aplicação estão diretamente relacionados com tempo de resposta, taxa de transferência e uso de recursos está diretamente relacionada com a experiência do usuário que durante grandes eventos ou grandes picos de acesso, tais como *black friday*, *cyber monday* e Natal tendem a formar a opinião sobre aplicações ou serviços de software.

## 2.2. LoadRunner

LoadRunner é uma aplicação voltada para testes de desempenho, atualmente é distribuída pela Microfocus e anteriormente pela HP (Hewlett-Packard). O LoadRunner possui a arquitetura distribuída em cinco sub-aplicações em cinco sub-aplicações, seus componentes e aplicações descritos na 1.

Tabela 1. Componentes da arquitetura do LoadRunner.

Componente	Responsabilidades
<i>Load Generator</i>	Responsável por gerar carga em aplicações utilizando os <i>scripts</i> criados no VuGen
<i>VuGen</i>	Gerador de usuários virtuais, responsável pela criação e manutenção dos <i>scripts</i>
<i>Controller</i>	Controla e Gerencia as instâncias da Load Generator, especificando a ordem de <i>scripts</i> que serão executados e por quanto tempo. Receptor do status de monitoramento das execuções em tempo real.
<i>Agent Process</i>	Agente responsável por iniciar as execuções em máquinas remotas e por gerenciar a conexão entre a Load Generator e a Controller.
<i>Analysis</i>	Com base nas informações recebidas das Load Generators, constrói os relatórios de monitoramento.

A escolha da *LoadRunner* como aplicação se deu por conta dos conectores da aplicação que se integram ao *SAP*, o que facilita a criação dos *scripts* e a integração com a massa de dados que é pré-configurada na aplicação e será utilizada de acordo com a configuração dos *scripts* e da *RunLogic* definida antes da execução dos *scripts*.

O *LoadRunner* possui uma arquitetura de fácil entendimento, em que existe um módulo central chamado de *Controller*. A *Controller* gerencia as instâncias da *Load Generator* que por sua vez tem a responsabilidade de gerar carga no sistema que será testado. A *Controller* é responsável por organizar a ordem de execução dos *scripts*, as credenciais que serão usadas nos testes e também o *Think-Time*, ou o tempo que *VUser* interage com o sistema e com base nesse tempo a subida dos novos usuários é definido. Além disso, a coleta de informações como por exemplo o tempo de resposta de cada transação e o *Throughput* (Taxa de Transferência).

## 3. Sistema em Teste

O *SAP* é atualmente o maior sistema integrado de gestão empresarial (*Enterprise Resource Planning - ERP*) existente no mercado. A empresa Alemã possui dentro da sua carteira de clientes 99 das 100 maiores empresas do mundo. O *SAP* atua em diversas frentes desde a compra de um novo produto, catalogação e manutenção. A análise desse artigo será direcionada as transações IW21 e IW22, diferente dos sistemas convencionais e dos testes de desempenho convencionais, as transações *SAP* tem ampla conexão com todos os módulos dentro do *SAP EC0* da empresa em que foi instalado, juntamente com todas as regras de negócio, o que aumenta a complexidade e o entendimento das diversas regras. Vale ressaltar que ambas as transações citadas anteriormente são *Standard*, logo já vem com o padrão de mercado adotado pela *SAP*. As transações IW21 e IW22 são responsáveis por criar novos planos de manutenção e editar planos de manutenção já existentes, que podem variar desde a movimentação de itens até ordens diversas.

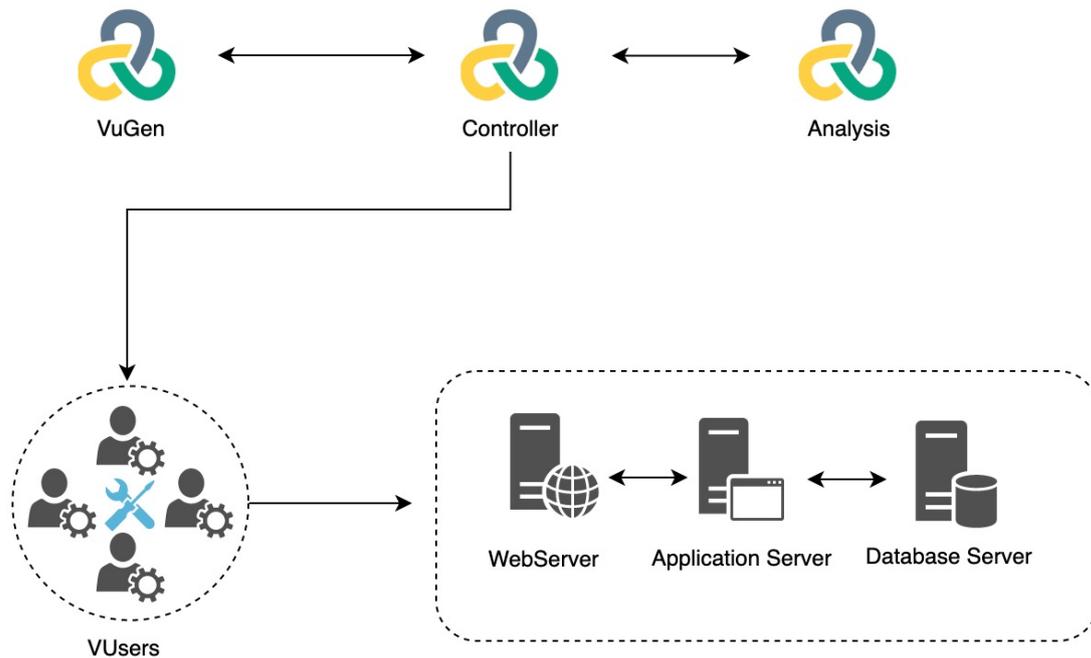


Figura 1. Arquitetura LoadRunner

## 4. Teste de Desempenho no SAP

### 4.1. Contexto

A empresa que está em análise neste estudo é uma das maiores mineradoras do mundo e tem atuado em diversos países, com a finalidade de tornar este estudo direcionado, iremos apresentar os testes realizados em 3 países diferentes, sendo eles: Brasil, Canadá e China. A necessidade deste teste em específico vem do volume de transações no porto de Ningbo-Zhoushan que, atualmente, segundo o SeaTrade [SeaTrade 2021] é o porto com o maior movimentação de cargas em tonelagem, tal valor aumenta a importância da comunicação entre múltiplas regiões solicitantes de informações ou de requisições de movimentação como por exemplo, compras e vendas.

No momento, os maiores ofensores do índice de impacto estão no tempo de resposta nas pesquisas e na catalogação de itens das transações *IW21 - Create PM Notification (General)* e *IW22 - Change Notification*, ambas responsáveis pela criação e alteração de notas de manutenção, as quais são responsáveis por catalogar as diversas movimentações dentro o porto em questão e outras diversas áreas. No contexto atual esta transação é utilizada múltiplas vezes em outros diversos módulos do SAP, o que aumenta drasticamente o uso do módulo de *Plant Maintenance*, permitindo ao usuário manter equipamentos e possibilitar o planejamento eficiente de novas implantações e migrações de carga.

## 4.2. Objetivo

O histórico já exemplificado nesta seção nos permitiu criar um plano de testes dividido em duas etapas distintas. Mas novamente pensando em criar um ambiente de testes espelhado, sem divergências na coleta dos tempos de respostas, estamos mantendo o mesmo método de teste com a subida dos usuários (*Ramp Up*) e com o monitoramento de um período de alta carga por tempo suficiente para coletar os indicadores (*Steady State*). O objetivo é atingir aproximadamente 1000 usuários simultâneos, para coletar o comportamento atual da aplicação como base para a 2ª bateria de testes.

A 1ª bateria de testes visa estimar o comportamento do teste de desempenho no ambiente atual, com 6 máquinas e tentar alcançar a métrica de 1000 usuários virtuais (*VUsers*) simultâneos.

**Tabela 2. Nome da tabela**

<b>Operação</b>	<b>Objetivos e Resultados-Chave</b>
<i>Ramp Up</i>	2 <i>VUser</i> a cada 3 segundos até o pico de 200 <i>VUsers</i> logados simultaneamente e operando na aplicação.
<i>Steady State</i>	Sistema permanece em estabilidade ( <i>Steady State</i> ) por 10 minutos.
2º <i>Ramp Up</i>	Após a estabilidade ( <i>Steady State</i> ) de 10 minutos o <i>Ramp Up</i> é retomado incluindo mais 100 <i>Vusers</i> .
2º <i>Steady State</i>	Seguimos com este processo até 1000 <i>VUsers</i> ou o sistema apresentar degradação significativa.

Dada a estratégia apresentada na Tabela 2, as operações de 2º *Ramp Up* e 2º *Steady State* serão repetidas múltiplas vezes até, o pré-requisito de 1000 *VUsers* simultâneos ou qualquer tipo de degradação, listada conforme segue:

- Aumento no tempo de resposta das transações;
- Existência e/ou recorrência de erros de acesso, como por exemplo: *400 Bad Request*, *404 Not Found*, *408 Request Timeout*;
- Atraso na subida de novos usuários únicos.

Já a 2ª bateria de testes com base nos resultados dos testes anteriores, visa avaliar o novo comportamento do ambiente buscando também 1000 *VUsers* simultâneos, mas avaliando as novas máquinas e a escalabilidade delas diante do número de usuários simultâneos.

## 4.3. Cenário de Teste

Para a execução do teste em questão será necessário criar cenários *end-to-end* para melhor compreensão de todo o tempo que um usuário real leva para realizar as operações desde o login no SAP EC0 até a geração do protocolo final de ambas as transações, tanto a IW21 quanto a IW22. A Tabela 3 descreve as etapas em sequência (Seq.) em que os *scripts* serão executados. Vale ressaltar que apesar de simplório, cada um dos *scripts* irá tocar em uma camada diferente do ambiente como um todo. O *script* de login irá coletar os tempos de resposta da autenticação no SAP EC0, o acesso a transação tanto IW21 quanto IW22 irá carregar o catálogo de informações das transações de *Plant Maintenance* juntamente com os registros de tempo de resposta, o que irá facilitar o entendimento dos ofensores.

**Tabela 3. Sequencia de operações das transações IW21 e IW22**

Seq.	Transação	Observação
#1	ALL_01_Login	<i>Script</i> destinado a realizar a entrada no SAP EC0 para todos os <i>VUsers</i> .
#2	IW21_02_Acessar_Transacao	Responsável por acessar a transação IW21 e coletar os tempos de resposta.
#3	IW21_04_Cria_Nota	Criação de notas na transação IW21 e coleta do tempo de resposta dos catálogos de produtos/serviços.
#4	IW22_02_Acessar_Transacao	Responsável por acessar a transação IW22 e coletar os tempos de resposta.
#5	IW22_03_Preencher_Dados	Preenchimento dos dados nas notas e novo registro das informações dos catálogos.
#6	IW22_04_Logout	Encerramento do acesso ao SAP EC0.

#### 4.4. Métricas e Indicadores de Desempenho

O processo de execução do *Ramp Up* teve a duração aproximada de 1h e 20 minutos. Assim, foi possível notar que próximo aos 10 primeiros minutos o ambiente começou a rejeitar as transações (registros vermelhos no gráfico), indicador esse que se repete e aumenta. Durante a execução foi realizado o diagnóstico das falhas de acesso, o que nos levou a entender o encavalamento dos acessos e mesmo com o grande volume de contas disponíveis para acesso, a fila de aceite de requisições começou a rejeitar requisições com tempo de espera (*time out*), com o intuito de proteger a estabilidade do ambiente como um todo.

Ao chegar no patamar de 580 usuários o ambiente iniciou clara degradação e atingindo o pico em 609. Então iniciamos a fase de *Steady State* e avaliação de estabilidade do sistema, em que o número de transações se manteve estável e operando, já os erros de acesso diminuíram drasticamente até o fim da primeira bateria de testes.

A media de tempo de resposta demonstra os principais itens ofensores durante a bateria de testes, e apesar da pequena oscilação em todas as *scripts*, o maior ofensor está na criação de notas na transação *IW21*, devido o grande número de pesquisas em catálogos que, consequentemente, aumenta o número de requisições de informações externas ao módulo de *Plant Maintenance*, o que gerou um pico de 651,066s (aproximadamente 10 minutos) onerando o ambiente (Tabela 4 e Figura 2). Em comparação com o número de *VUsers* é possível notar que o aumento drástico no tempo de resposta acompanhou a rejeição de novos *VUsers*.

Durante as baterias de testes foi possível encontrar a transação responsável pela maior média de tempo de resposta. O tempo elevado ocorreu devido ao excesso de pesquisas do catálogo de informações, em que acontece a conexão com outras diversas transações SAP e consome o maior volume de dados dentro dos *scripts*. A criação de novas notas fiscais durou em média 26,9 segundos (Tabela 4). O tempo de operação de cada um dos *scripts* foi configurado como dinâmico, assim como o espaçamento de subida de novos *VUsers*.

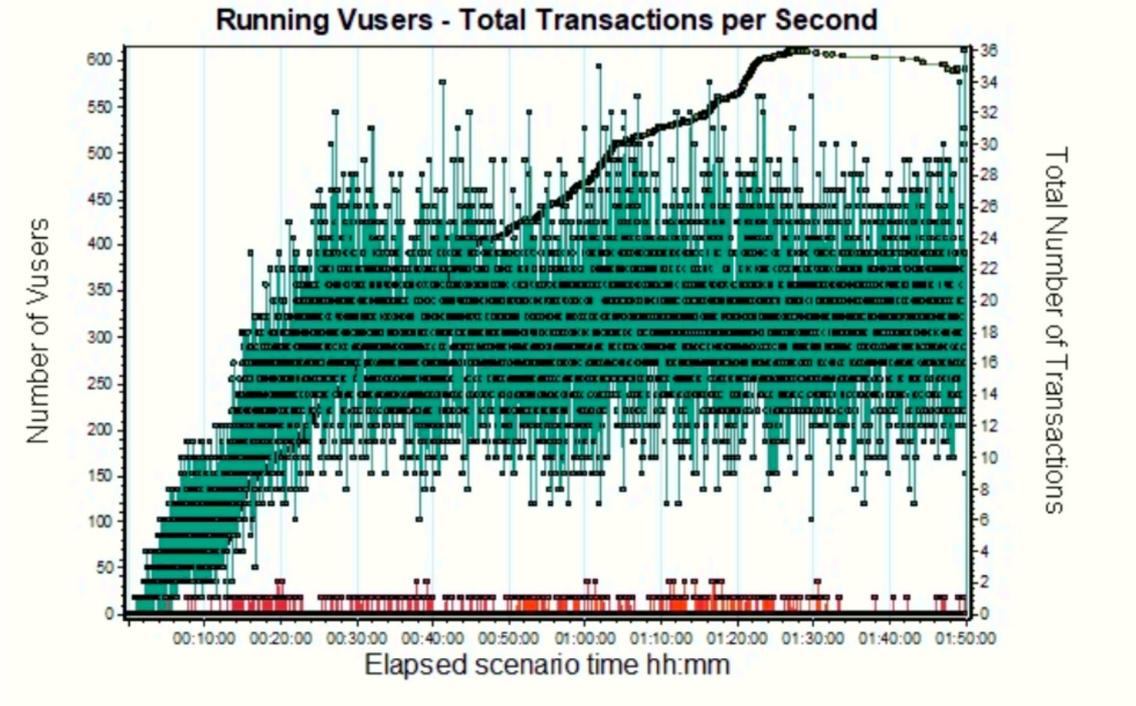


Figura 2. Ramp up dos usuários virtuais em execução

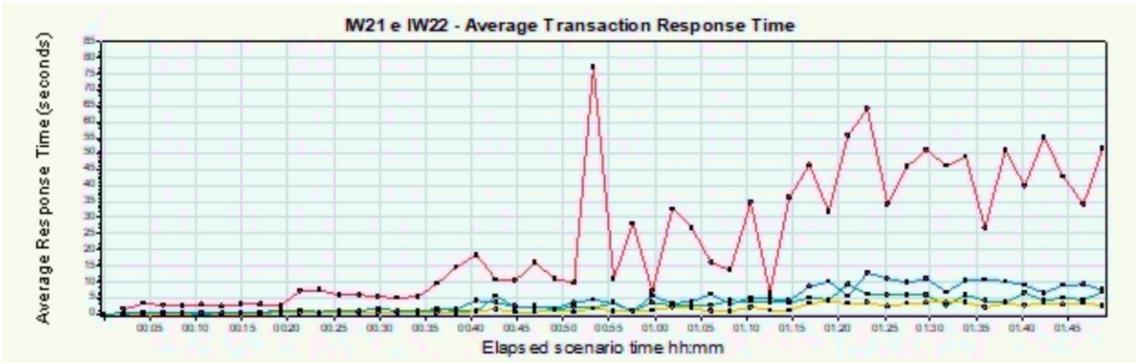


Figura 3. Tempo médio de resposta da transação

**Tabela 4. Estatísticas básicas do tempo de resposta**

Color	Escala	Transação	Mínimo	Médio	Máximo
	1	IW21_02_Acessar_Transacao	0,039	3,161	61,187
	1	IW21_04_Cria_Nota	1,105	26,871	651,066
	1	IW22_02_Acessar_Transacao	0,036	1,789	18,556
	1	IW22_03_Preencher_Dados	0,162	5,241	62,874

**Tabela 5. Configuração dos testes das transações IW21 e IW22**

Transações	Usuários Concorrentes	Volume Médio de Transações por Hora	Think Time	Pacing
IW21_IW22	20	1256	Randômico	Randômico

#### 4.5. Análise dos Resultados e Discussões

Todo o processo de execução exigiu vasta análise e comparação dos resultados como um todo. Os testes apresentaram evidências de que é possível encontrar relação entre o aumento no tempo de resposta e a rejeição na subida de novos *VUsers*. Durante toda a bateria de testes foi possível avaliar junto ao time de *middleware* e verificar se os dados apresentados na *Controller* refletiam no ambiente SAP. Assim, como resultado foi possível avaliar uma variação de 2 a 3 usuários operando simultaneamente, o que foi considerado dentro da margem de erros e da alta quantidade de *VUsers* em execução. Consideramos também avaliar as notas geradas durante todo o período de execução, bem como o baixo tempo de resposta no preenchimento dos dados, juntamente com o diagnóstico do time de operações SAP. Desta forma, entendemos que não houve degradação na aplicação e todas as notas que foram gravadas no sistema foram protocoladas com sucesso.

#### 5. Considerações Finais

Os pontos de ação demonstrados são Indicadores e chaves de desempenho que deverão ser pontuadas nos próximos passos desse estudo, o catálogo de informações da transação IW21, por exemplo tem relação direta com os pontos de melhoria que devem ser avaliados após a migração do ambiente atual, visto que a transação possui o maior tempo de resposta entre todos os *scripts* avaliados. Apesar dos pontos citados anteriormente, entendemos que o ambiente é sólido, confiável e apesar dos problemas gerados pela carga de usuários utilizando simultaneamente o processo, as operações não foram comprometidos, os dados permaneceram seguros na aplicação mesmo com aproximadamente 1256 usuários operando duas transações do mesmo modulo em um longo período de tempo. Dito isto, os próximos passos irão caminhar passo a passo com a *User Experience*, graças a alta confiabilidade sera possível focar nas condições reais e na interação entre as transações e não apenas na sobrecarga da aplicação.

#### Referências

SeaTrade (2021). Ningbo-zhoushan world's top cargo handling port in 2021. In SeaTrade, editor, *Ningbo-Zhoushan world's top cargo handling port in 2021*. SeaTrade.

Sommerville, I. (2019). *Engenharia de Software*. Pearson Brasil.

Weiss, C., Westermann, D., Heger, C., and Moser, M. (2013). Systematic performance evaluation based on tailored benchmark applications. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering, ICPE'13*, pages 411–420, New York, NY, USA. ACM.