

UMA PROPOSTA PARA A ANÁLISE DO MAKESPAN E DO DESEMPENHO DO MODELO DE EXECUÇÃO BASEADO EM PROCESSOS NO CONTEXTO DE INTEGRAÇÃO DE APLICAÇÕES¹

A PROPOSAL FOR MAKESPAN AND PERFORMANCE ANALYSIS OF THE PROCESS-BASED EXECUTION IN THE CONTEXT OF APPLICATION INTEGRATION

Carolina Hilda Schleger², Sandro Sawicki³, Rafael Zancan Frantz⁴

¹ Projeto de Pesquisa realizado no programa de Pós-Graduação em Modelagem Matemática e Computacional da UNIJUÍ.

² Mestranda do Programa de Pós-Graduação em Modelagem Matemática e Computacional da UNIJUÍ.

³ Professor do Programa de Pós-Graduação em Modelagem Matemática e Computacional da UNIJUÍ.

⁴ Professor do Programa de Pós-Graduação em Modelagem Matemática e Computacional da UNIJUÍ.

RESUMO

Para a realização de um processo de integração é importante definir um motor de execução, o qual pode ser implementado em dois modelos: task-based e process-based. Nos dois modelos o motor de execução recebe informações em forma de mensagens, as quais são processadas ao longo do tempo por meio de recursos computacionais denominados *threads*. Este artigo aborda o comportamento do motor de execução do processo de integração baseado no modelo process-based. O objetivo é analisar o comportamento do processo de integração quando as mensagens são processadas diante de diferentes taxas de entrada, variando o número de *threads* e o *makespan*. O *makespan* é uma métrica de desempenho que calcula o tempo que uma mensagem leva para ser processada dentro de um processo de integração. Compreender o comportamento do motor de execução pode auxiliar o engenheiro de software a determinar ações para aumentar o processamento de mensagem em menor tempo de execução. Um motor de execução eficiente é fundamental devido à crescente demanda por informações e a alta conectividade presente na vida cotidiana das pessoas e das organizações.

Palavras-chave: Integração de Aplicações. Modelo de Execução baseado em Processos. Mensagens processadas. *Makespan*.

ABSTRACT

To carry out an integration process, it is important to define an execution engine, which can be implemented in two models: task-based and process-based. In both models, the execution engine receives information in the form of messages, which are processed over time through computational resources called threads. This article discusses the behavior of the execution engine of the integration process based on the process-based model. The objective is to analyze the behavior of the integration process when messages are processed with different



input rates, varying the number of threads and the makespan. The makespan is a performance metric that calculates the time it takes for a message to be processed within an integration process. Understanding the behavior of the execution engine can help the software engineer to determine actions to increase message processing in less execution time. An efficient execution engine is essential due to the growing demand for information and the high connectivity present in the daily life of people and organizations.

Keywords: Application Integration. Process-based model. Messages processed. Makespan.

INTRODUÇÃO

A geração de conhecimento e de funcionalidades nos diversos ramos da vida caracterizam uma sociedade atual que gera e armazena uma vasta quantidade de dados. Estes dados apresentam informações relacionadas a, por exemplo, culturas sociais, financeiras, educacionais, judiciais e de saúde, podendo ser armazenados em arquivos de texto, planilhas, imagens, áudio e vídeos (GANDOMI, 2015). Características que estão presentes no contexto de *Big Data* que é marcado pelos 5 Vs: volume, velocidade, variedade, veracidade e valor (GANDOMI, 2015; LANEY, 2001). Muitas dessas características estão presentes nos desafios das empresas ou em organizações, em integrar informações de dados e funcionalidades entre os aplicativos de software que não tiveram sua criação projetada para o compartilhamento de funcionalidades. Para solucionar estes fatos a Integração de Aplicações Empresariais (EAI) elaboram metodologias, técnicas e ferramentas para projetar, implementar, executar e monitorar o processo de integração. Para executar o plano de ação, ou seja, os códigos com as ordens de como realizar a comunicação e funcionalidade de diferentes aplicativos, a plataforma de integração necessita de uma linguagem específica de domínio, um conjunto de ferramentas de desenvolvimento, um ambiente teste, uma ferramenta de monitoramento e, um sistema de tempo de execução (FRANTZ, 2011; RITTER, 2017). Para a realização do processo integração é importante definir um motor de execução, o qual pode ser implementado em dois modelos: *task-based* e *process-based*. Nos dois modelos o motor de execução recebe informações em forma de mensagens, as quais denominamos como a taxa de mensagens que podem variar de acordo com o tempo. Para processar as mensagens são utilizados os recursos computacionais chamados de *threads*, as quais estão armazenadas dentro de um *pool* de *threads*. A diferença entre os dois modelos está na organização da



utilização das *threads*. O modelo *task-based* direciona cada *threads* para uma tarefa e quando este finaliza sua parte volta para o *pool* global, não acompanhando a realização das orientações de cada mensagem até o final do processo de integração; assim, mais de uma *threads* auxilia no processamento de uma mensagem. Já quando uma mensagem entra no processo de integração utilizando o modelo *process-based*, a *thread* a acompanha até que ela seja finalizada, sendo criadas instâncias simultâneas para processar as demais mensagens que são enviadas ao processo (HOHPE, 2004). Esta pesquisa se concentra em estudar o comportamento do motor de execução do processo de integração baseado no modelo *process-based*. Com o objetivo de analisar o comportamento do processo de integração quanto a mensagens processadas diante de diferentes taxas de entrada de mensagens, variando o número de *threads* e o *makespan*. O estudo do comportamento do motor de execução é importante para auxiliar o engenheiro de software a determinar ações que o façam processar o máximo de mensagem possíveis em menor tempo. Este fator tem sua importância principalmente pelo fato de as informações de diferentes contextos estarem em constante mudanças ou serem criadas a cada instante, envolvendo o contexto de *Big Data*. Que necessita de respostas seguras e confiáveis em menor tempo possível.

Nas seções seguintes é abordado a metodologia, a análise dos dados referente ao *makespan* e o número de mensagens processadas por segundo e considerações finais.

METODOLOGIA

A metodologia define a organização dos caminhos que serão percorridos para desenvolver uma pesquisa (FONSECA, 2002). Ela é importante ser estabelecida no início, pois auxilia na definição do método científico, o objetivo e o procedimento técnico a ser utilizado. Apresentando um discurso lógico, sistemático, bem-argumentado sobre seu tema e objetivo; distanciando-se do conhecimento embasado no senso comum e ideológico (PRODANOV, 2013). A pesquisa de natureza aplicada se classifica pelo método hipotético dedutivo, como objetivo de uma pesquisa exploratória com procedimento bibliográfico, experimental e estatístico. A metodologia de pesquisa bibliográfica está de acordo com o objetivo de analisar os métodos estatísticos considerados robustos perante conjuntos de dados



oriundos do contexto de Big Data e o funcionamento do motor de execução *process-based*. A revisão bibliográfica tem como intuito agregar todas as evidências existentes em uma pergunta e, também, para apoiar o desenvolvimento de diretrizes baseadas em evidências para o pesquisador (KITCHENHAM, 2009). O conhecimento construído pelas pesquisas anteriores possibilita a construção de hipóteses de que podem ser confirmadas ou rejeitadas. O procedimento de pesquisa experimental, assim como o estatístico, teve o intuito de investigar a influência dos recursos computacionais sobre o desempenho do motor de execução *process-based*. Analisando o impacto da variação do número de *threads* sobre diferentes taxas de entradas de mensagens. Voltando-se principalmente para o número de mensagens processadas e o tempo de execução de cada uma, ou seja, o *makespan*. A pesquisa experimental evidencia o estudo do comportamento das variáveis relacionadas com o objeto, verificando os possíveis motivos de determinado fato (PRODANOV, 2013). Por apresentar estas características o presente estudo tem uma abordagem quantitativa.

EXPERIMENTO

O experimento deste trabalho é fruto de uma pesquisa desenvolvida no grupo GCA. Baseado no método hipotético dedutivo exploratório de forma quantitativa, possibilitando a avaliação de hipóteses e suas respostas (PRODANOV, 2013). Realizado com o intuito de avaliar o comportamento do motor de execução baseado no modelo *process-based* implementado na plataforma de integração Camel. Em uma máquina equipada com 16 processadores Intel Xeon CPU E5-4610 V4, 1.8GHz, 32 GB ou RAM. Em um sistema operacional Windows Server 2016 Datacenter de 64 bits, apresentado JAVA versão 8.0 com atualização 152 e executado na Plataforma Camel. Com capacidade de memória de até 24 GB. Sendo que para sua execução a máquina foi desconectada da internet e não havia outro software instalado nela.

Estabeleceu-se duas questões de pesquisa para nortear este estudo.

RQ1: Qual o impacto do número de *threads* no *pool* global em relação ao desempenho do sistema de tempo de execução sobre as mensagens que implementam o modelo de execução *process-based*?



RQ2: Como o número de *threads* no *pool* global e a taxa de mensagens de entrada afetam o *makespan* de um processo de integração no modelo de execução *process-based*?

A partir destas duas questões, foram estipuladas duas hipóteses que por meio do experimento serão confirmadas ou refutadas:

H1: A hipótese estabelecida para a primeira questão é de que à medida que é incrementado o número de *threads* o sistema consiga processar todas as taxas de mensagem injetadas, ou seja, que o processamento seja constante em um comportamento proporcional linear.

H2: Em relação à hipótese para a segunda questão é que de acordo com o aumento do número de *threads* disponíveis há a diminuição do tempo de processamento de mensagens, ou seja, do *makespan*.

Para a realização do experimento escolheu-se o modelo de solução de integração Coffee Shop conforme as orientações de Hohpe et al. (2004). A Figura 1 ilustra os caminhos desenvolvidos por este modelo de solução de integração, que inicia seu fluxo de trabalho quando um cliente realiza um pedido de bebidas ao caixa e este, anota o pedido adicionando à fila de atendimento. O pedido realizado pode apresentar a solicitação de bebidas quentes e frias. Ao sair da fila de atendimento, o pedido é desmembrado conforme a solicitação do cliente, em que as bebidas quentes serão atendidas pelo barista de bebidas quentes e as bebidas frias pelo barista de bebidas frias. Não havendo interação de um barista com outro barista. Finalizado a bebida pelo barista, este é colocado em uma bandeja identificada e levada pelo garçom até o cliente novamente, finalizando o processo. Com a divisão dos pedidos, cada pedido e bebida é identificada ao seu pedido correspondente, sendo entregue apenas quando as solicitações do cliente estiverem finalizadas. Enquanto isso, o caixa pode continuar a adicionar pedidos a fila de pedidos, o que pode acarretar em uma sobrecarga no sistema pelo motivo dos baristas não terem capacidade de visualizar a dimensão dos pedidos realizados. Ou seja, não há relação entre o caixa e os baristas além da fila de pedidos. Este modelo exibe duas formas de interações: síncronas e assíncronas. Síncrona refere-se a uma ordenação não modificável que é o momento onde o cliente realiza

seu pedido ao caixa. Assíncrona está representada no poder dos baristas entregar os pedidos solicitados em uma ordenação distinta em que foram feitos, otimizando o processo e o tempo.

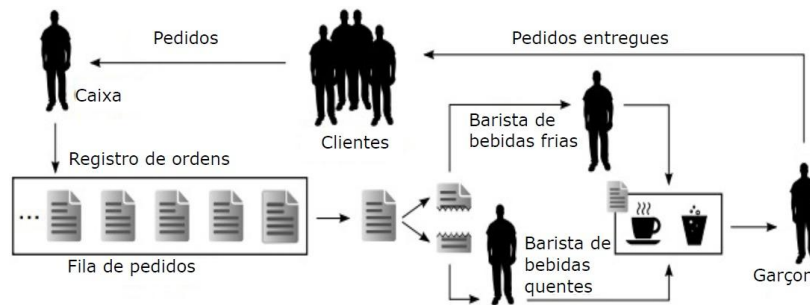


Figura 1: Ilustração do processo de integração da *Coffee Shop*.

No estudo experimental foram executados 1100 (1 x 20 x 55) experimentos. Cada experimento foi realizado durante 60 segundos, apresentado uma taxa de entrada de mensagens e usado uma quantidade de *threads*, esta mesma ação com os mesmos valores definidos repetiu-se 25 vezes. Por exemplo, um experimento realizado durante o tempo de 60 segundos, injetando uma taxa de entrada de 500 mensagens por segundo em um processo de integração utilizando apenas uma *threads*. A repetição de um experimento é necessária para verificar alguma regularidade no comportamento do objeto de estudo. Esta regularidade possibilita ao pesquisador definir parâmetros, explicar alguns fatos e realizar uma suposição apoiada em argumentos com alguma margem de erro. Desta forma, os dados apresentados para análise são oriundos de uma média aritmética das 25 repetições. As mensagens inseridas no processo de integração possuem um cabeçalho apresentando informações principais pré estabelecidas e em seu corpo os dados que serão analisados no processo de integração e um arquivo em formato XML. Para registrar os dados de *makespan* e a quantidade de mensagens processadas adicionou-se algumas configurações no código fonte do modelo de solução de integração *Coffee Shop*. Ao final do processo estas informações foram armazenadas em um arquivo de texto, transportando em um planilha para serem analisadas. Para calcular o *makespan* de cada mensagem configurou-se para que o cabeçalho anotasse o momento de entrada da mensagem no processo de integração e sua saída. A diferença entre o tempo de



saída em relação à entrada determina o valor do *makespan* de cada mensagem processada. E neste mesmo código fonte estabeleceu-se um contador do número de mensagens inseridas no processo de integração e ao final do tempo de 60 segundos quantas haviam sido processadas ao total.

MAKESPAN

A análise prévia do experimento foi realizada a partir do olhar da média das vinte cinco repetições, gerando a discussão dos resultados dos dados a seguir. Todos gráficos analisados nesta seção demonstram o comportamento do *makespan* em relação ao incremento de *threads* para diferentes taxas de entrada de mensagens. O *makespan* é o tempo em que uma mensagem necessita para ser executada, ou seja, a diferença de tempo entre o momento de entrada com a saída da mensagem. A Figura 2 apresenta o *makespan* em função do número de *threads* disponíveis para diferentes taxas de entrada de mensagem. Apresentando a variação da taxa de entrada de mensagens de 500 a 10000 msg/s, com variação numérica de 500 em 500. Analisando o comportamento do primeiro gráfico, verificou-se que este apresenta um comportamento exponencial decrescente. Visualizando que a diminuição de *makespan* está diretamente relacionado com a capacidade do processo de integração em processar mais mensagens por segundo de acordo com o aumento do número de *threads*. Sendo que em taxas mais elevadas a diminuição do *makespan* é menor, pois o sistema não consegue finalizar todas as mensagens que entram. Apresentado, também, uma estabilidade de acordo com o valor das variáveis envolvidas. Na mesma Figura 2, visualiza-se no segundo gráfico que o incremento acima de 50 *threads*, independente da taxa de entrada de mensagens, o sistema não apresenta a redução no tempo de execução. E sim, houve um leve aumento do *makespan* que é possível visualizar. O segundo gráfico da Figura 2 evidencia o período do incremento de 50, 100, 150, 200, 250 e 300 *threads* no processo de integração. O aumento do *makespan* neste período específico é possível afirmar, pois nas taxas acima de 2500 msg/s não apresenta uma reta constante e sim, irregular. Uma explicação para este fator, é que o incremento do número de *threads* sobrecarrega o gerenciamento de recursos do processo de integração, fazendo este,



destinar mais tempo para organizar os recursos disponíveis do que efetuar as mensagens que entram.

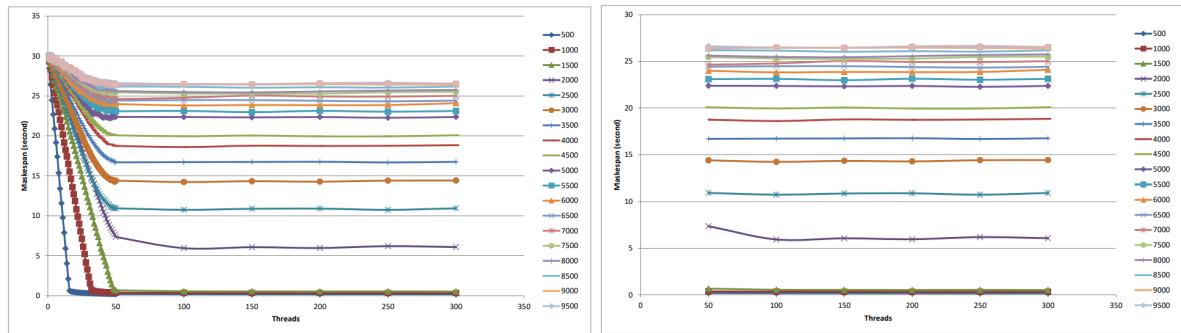


Figura 2: Makespan em função do número de *threads* no intervalo de 500 a 10000 msg/s.

Os dados do gráfico da Figura 3 descrevem o *makespan* em função do número de até 50 *threads* disponíveis para diferentes taxas de entrada de mensagem. Observamos que o *makespan* tende à zero para as taxas de mensagens de 500, 1000 e 1500 msg/s, pois o sistema é capaz de processá-las completamente. Para taxas mais elevadas este fator não ocorre, pois o sistema não consegue processar todas as mensagens inseridas, por mais que se acrescente *threads*.

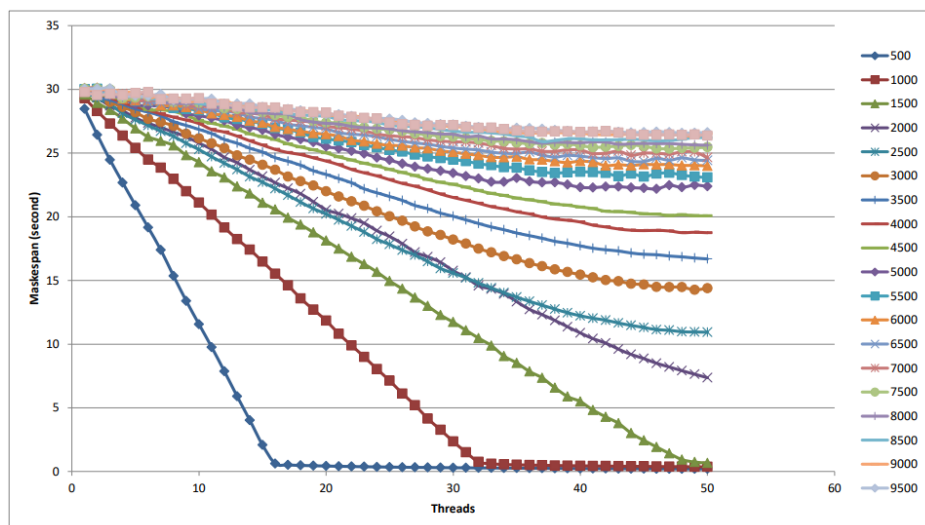


Figura 3: Makespan em função do número de *threads* no intervalo de 500 a 10000 msg/s.



MENSAGENS PROCESSADAS

Nesta seção é discutida a análise dos dados, da média das vinte cinco repetições, sobre a perspectiva da segunda hipótese sobre a capacidade do motor de execução processar mais mensagens por segundo. Todos os gráficos a seguir descrevem o número de mensagens processadas com o incremento de *threads* para diferentes taxas de entrada.

A Figura 4 apresenta o número de mensagens processadas por segundo de acordo com o número de *threads* disponíveis, com a variação da taxa de entrada de mensagens de 500 a 10000 msg/s. No contexto em geral, de acordo com o aumento de *threads* disponíveis para o processo de integração, percebe-se um aumento do processamento de mensagens por segundo até um certo ponto onde, então, evidencia-se uma estabilidade do sistema, descrevendo uma trajetória logarítmica crescente. Este comportamento pode ser explicado, por exemplo, com uma taxa de entrada de 1500 msg/s, de acordo com o acréscimo de número de *threads* no processo, este é capaz de processar mais mensagens por minuto. Entretanto, Figura 5, percebe-se que após o acréscimo de 44 *threads* ao processo de integração não apresenta a mesma taxa de crescimento de mensagens processadas e, sim, uma estabilidade. Significando, que o acréscimo de *threads* não influencia com a mesma intensidade como inicialmente. Esta estabilidade tem sua variação de acordo com o valor da taxa de entrada e o número de *threads* utilizados.

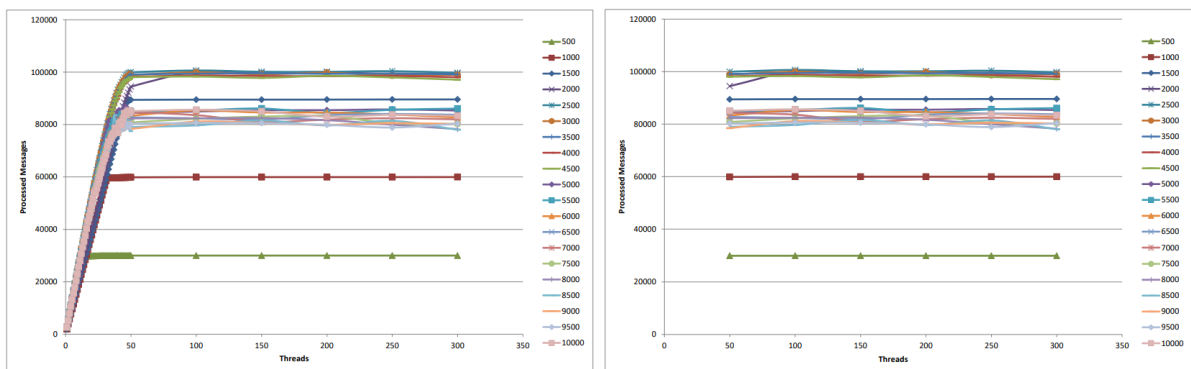


Figura 4: Número de mensagens processadas no intervalo de 500 a 10000 msg/s em função do número de *threads*.

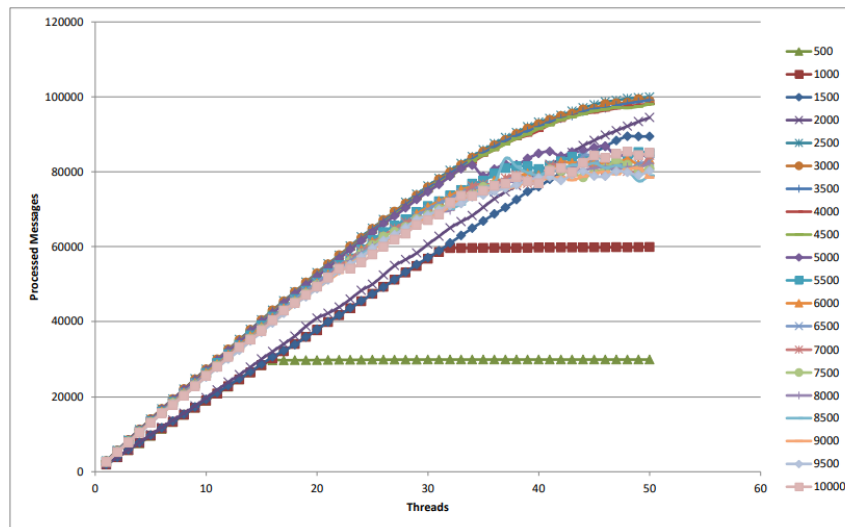


Figura 5: Número de mensagens processadas no intervalo de 500 a 10000 msg/s em função do número de até 50 threads.

Entretanto, percebe-se que o comportamento do número de mensagens processadas quando há o incremento maior que 50 threads, no segundo gráfico da Figura 4, há uma leve queda no processamento das taxas de entrada de mensagem. Desta forma, não é possível afirmar que o incremento de threads auxilia sempre no processamento e, sim, que este pode prejudicar no desempenho da integração.

CONSIDERAÇÕES FINAIS

O experimento demonstrou que o processo de integração por meio do motor de execução *process-based* é eficiente, capaz de processar todas as mensagens inseridas, quando o intervalo de taxa de entrada de mensagem está entre 500 a 2000 msg/s. Em que, apresenta o processamento de todas as mensagens inseridas e o *makespan* tende a zero segundo. Lembrando que o *makespan* sempre terá um valor de tempo para uma mensagem ser executada, que é a diferença entre o momento de entrada e saída.

Quando a taxa de entrada varia de 2500 a 4500 msg/s, o processamento de mensagens não é totalmente efetivado, mesmo com o incremento do número de threads. Percebeu-se que o início do incremento de threads aumenta o processamento de mensagem, no entanto, o aumento não controlado de threads pode prejudicar no desempenho do motor de



execução. Uma explicação para este efeito pode ser devido à arquitetura de processos em que demanda mais recursos computacionais como a memória devido a alocação e desalocação de *threads*.

O intervalo igual ou maior que 5000 msg/s, o processamento de mensagem apresenta a elevação do *makespan* principalmente pelo fato de o número de mensagens que saem do processo é menor que o número de mensagens injetadas, indicando uma ineficiência no processamento para as taxas de 9000, 9500 e 10000 msg/s.

REFERÊNCIAS BIBLIOGRÁFICAS

FONSECA, J. J. S. da. *Apostila de metodologia da pesquisa científica*. João José Saraiva da Fonseca, 2002.

FRANTZ, F. Z.; QUINTERO, A. M. R.; CORCHUELO, R. A Domain-Specific Language to Design Enterprise Application Integration Solutions. *Int. J. Cooperative Inf. Syst.*, v.20, p. 143 - 176, 2011.

FREITAS, C. de. *Metodologia do trabalho científico: Métodos e técnicas da pesquisa e do trabalho acadêmico* - 2 a edição. Editora Feevale, 2013.

GANDOMI, A.; HAIDER, M. *Beyond the hype: Big data concepts, methods, and analytics*. *International Journal of Information Management*, v. 35, n. 2, p. 137–144, 2015. ISSN 0268-4012. Disponível em:

<<https://www.sciencedirect.com/science/article/pii/S0268401214001066>>.

HOHPE, G.; WOOLF, B.; BROWN K.; CRUPI, J.; FOWLER, M. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley, 2004. ISBN 0-321-20068-3.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, v. 51, n. 1, p. 7–15, 2009. ISSN 0950-5849. Special Section - Most Cited Articles in 2002 and Regular Research Papers. Disponível em:

<<https://www.sciencedirect.com/science/article/pii/S0950584908001390>>.



LANEY, D. 3D. *Data Management: Controlling Data Volume, Velocity, and Variety*. [S.l.], 2001. Disponível em:

<<http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>>.

PRODANOV, C. C.; FREITAS, E. C. *Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico*. ISBN 8577171582, 20139788577171583, Ed. Feevale, 2013. Disponível em:

<<https://books.google.com.br/books?id=zUDsAQAAQBA>>.

RITTER, D.; MAY, N.; RINDERLE-MA, S. Patterns for emerging application integration scenarios: A survey, *Information Systems*. In: *Information Systems*. v. 67, p. 36 - 57, 2017. Disponível em: <<https://doi.org/10.1016/j.is.2017.03.003>>.