

Análise de desempenho da Solução de Integração Café na Plataforma Apache Camel em Ambiente Paralelo

Matheus H. Rehbein¹, Edinaldo Gaspar Da Silva¹, Rafael Z. Frantz¹,
Edson Luiz Padoin¹

¹Departamento de Ciências Exatas e da Terra
Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUI)
Ijuí – RS – Brazil

{mrehbein45, edigaspar01}@gmail.com {rzfrantz, padoin}@unijui.edu.br

Resumo. *A integração de aplicações apresenta demanda computacional e necessita de implementação paralela para atender as requisições. Neste trabalho apresenta-se uma análise de desempenho de uma implementação paralela variando-se a quantidade threads e de mensagens para um ambiente de integração. Os testes demonstraram que o emprego de 64 threads apresentou o maior desempenho, e a maior taxa de gargalos foi alcançada com 6400 msg/s.*

1. Introdução

A integração de aplicações é utilizada para realizar a comunicação entre aplicações [Hohpe and Woolf 2004] presentes dentro de um ecossistema de software heterogêneo [Messerschmitt et al. 2005]. A integração permite que aplicações desenvolvidas em épocas, empresas e tecnologias diferentes possam comunicar-se. Para auxiliar o desenvolvimento de integrações existem diferentes ferramentas de integração, sendo que todas buscam abstrair os processos complexos de uma integração, utilizando políticas de envio e recebimento de mensagens entre as aplicações.

As ferramentas de integração permitem que as mensagens possam transitar de forma concorrente, no entanto, as atuais ferramentas de integração não são capazes de definir a quantidade de *threads* ideal para cada ambiente, ficando esta especificação ao desenvolvedor. Esta funcionalidade é essencial no novo contexto de *Big Data*, onde uma grande quantidade de requisições são demandadas. Entretanto, um simples aumento na quantidade de *threads* pode acarretar em overhead e gargalos para o sistema como um todo. Assim sendo, faz-se necessário um fino ajuste deste parâmetro para cada ambiente.

Neste contexto, este trabalho tem o objeto analisar e identificar a quantidade ideal de *threads* para algumas soluções de integração. Para tanto, foram realizadas experimentações utilizando as plataformas de integração *Apache Camel* utilizando o problema de integração do *Café*. O restante do trabalho está organizado da seguinte forma: a Seção 2 discute os trabalhos relacionados. A Seção 4 apresenta a metodologia empregada nos testes, seguida da Seção 5 que apresenta os resultados alcançados. Por fim, A Seção 6 destaca as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

No trabalho de [Frantz et al. 2015] é apresentada uma metodologia para ranquear ferramentas de integração baseadas em suas métricas de manutenibilidade. Nele também é

apresentado um estudo de caso das plataformas de integração disponíveis utilizando alguns critérios. Neste estudo, o Guaraná ocupa a primeira posição, sem entretanto, uma análise de desempenho.

No contexto de integração de aplicações a computação em nuvem representa um novo paradigma no qual recursos como armazenamento e processamento podem ser acessados através da Internet e consumidos sob demanda. Sob este modelo de computação novos modelos de integração podem ser propostos, reduzindo o custo das empresas em infraestrutura [Zhang et al. 2010]. Nesse contexto, em [Frantz et al. 2011] é apresentado um modelo do motor para integração de aplicações em *Cloud* comparando o ambiente Apache Camel versão 2.7.2 e Guaraná RT versão 1.3 utilizando a solução de integração Café.

3. Solução Café

A Solução Café foi proposta por [Hohpe 2005] para o problema de um barista é apresentada na Figura 1 utilizando Guaraná DSL. O café é uma solução de *benchmark* conhecida na literatura de integração de aplicações, que consiste em simular as atividades de um barista. Nela os pedidos são inseridos na solução de integração pela porta de entrada denominada *Orders* e posteriormente encaminhados a tarefa *T1* do tipo *Chopper* que dividirá o pedido original em dois arquivos distintos, um contendo bebidas quentes e outro geladas.

Passada a primeira etapa, os arquivos gerados são encaminhados para a tarefa *T2* do tipo *Dispatcher*, que encaminhará os pedidos para os seus respectivos balcões. O processo a partir desse ponto é espelhado, logo, será abordado somente o fluxo das bebidas frias. Após passar pela tarefa *T2* o arquivo é encaminhado a tarefa *T3* do tipo *Replicator*, tal tarefa gerará duas cópias do arquivo por ela recebido, uma será encaminhada a tarefa *T5*, *Translator* e a outra à tarefa *T7*, *Correlator*.

Tarefas do tipo *Translator* aplicam modificações no arquivo recebido de maneira a criar um novo arquivo que possa ser lido pela aplicação destino, que no modelo é chamada de *Barista Cold Drinks*, que será responsável pela preparação do pedido do cliente. Tarefas *Correlator* fazem com que a cópia da mensagem só seja encaminhada a tarefa seguinte no momento que *Correlator* receber duas ou mais mensagens com o mesmo identificador. A tarefa seguinte *T9* é um *Context Based Enricher* que juntará os dois arquivos recebidos em *T7* em um único arquivo. Que será posteriormente encaminhado a tarefa *T11* que faz alguma coisa, e encaminha para tarefa *T12* do tipo *Agregator* que junta ambos as partes de um único pedido, se na tarefa *T1* havia sido separado o pedido em bebidas quentes e frias, em *T12* é realizada a junção do pedido novamente e encaminhado ao garçom.

4. Metodologia

Os testes foram realizados em um processador Intel Core i3-7100 3,90 GHz com 2 core, 4 threads e 3 MB de memória cache. Cada teste foi realizado 5 vezes, sendo apresentado a média destas execuções. Foram definidas taxas de entrada de 100, 400, 1600 e 6400 mensagens por segundos, testadas com 2, 4, 8, 16, 32 e 64 *threads*.

As execuções tiveram o seguinte fluxo: i) criação as mensagens, ii) inicialização da integração, iii) envio das mensagens, iv) processamento da integração, v) parada do

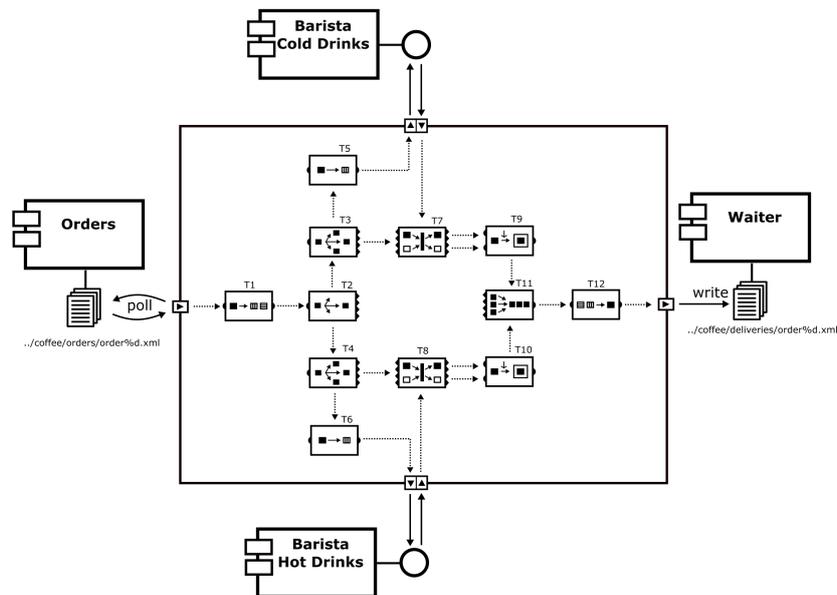


Figura 1. Proposta de solução de Integração do Café

envio das mensagens, vi) parada no processo de integração, vii) cálculo dos tempos, e viii) limpeza da memória na JVM. O processo de integração foi executado por 60 segundos.

5. Resultados

Na Figura 2 é apresentada a quantidade de mensagens processadas por minuto variando-se a quantidade de *threads* e a taxa de entrada de mensagens por segundo. Os demonstram que os menores desempenhos foram obtidos com 2 threads com a taxa de 6400 msg/s. Por outro lado, os maiores desempenhos foram alcançados quando empregado 64 threads e uma taxa de 1600 msg/s.

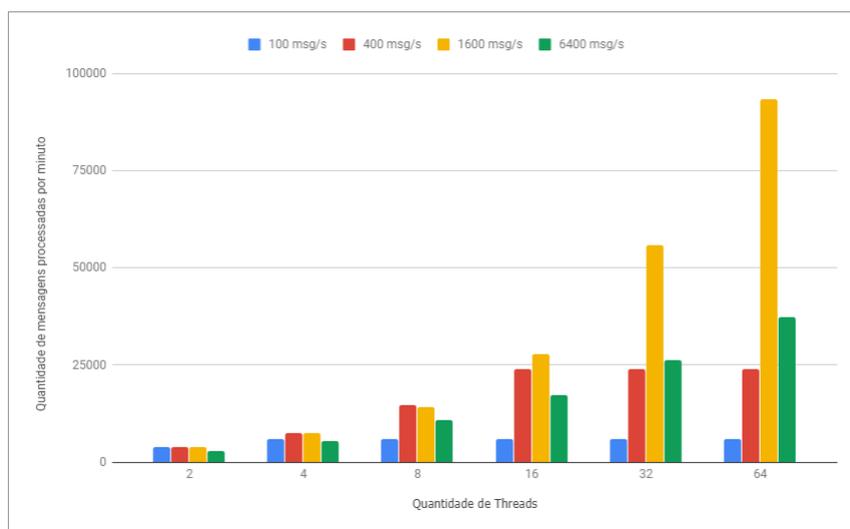


Figura 2. Quantidade de mensagens processadas por minuto

Percebe-se que a utilização de mais de 2 threads com a taxa de entrada de 100 msg/s apresentam resultados semelhantes. Nestes testes foram processadas todas as 6000

mensagens enviadas. O mesmo acontece para as mensagens processadas por mais de 16 threads com a taxa de 400 msg/s, processando todas as 24000.

A utilização da taxa de 6400 msg/s causou redução da quantidade de mensagens processadas. Isso ocorre devido a grande quantidade de mensagens sendo injetadas pela solução, o que resulta em gargalos de entrada acarretando a não leitura de muitas mensagens.

Foi também verificado essa característica em todas as quantidades de *threads* testadas. No entanto, uma redução menos significativa da quantidade de mensagens processadas foi verificada com 4 e 8 threads.

6. Conclusão e Trabalhos Futuros

Neste trabalho foram apresentados resultados das experimentações utilizando a solução de integração do Café na plataforma de integração Apache Camel. Através dos testes realizados foi verificado a presença de gargalos quando é aumentado a quantidade de mensagens enviadas. Também, foi identificado um aumento da quantidade de mensagens processadas quando um maior número de threads é empregado.

Os resultados apontam que um aumento da taxa de mensagens enviadas por segundo, causa gargalos a partir de 1600 msg/s. Entretanto, este problema é sanada com um aumento da quantidade de *threads*. Por outro lado, quando testado o ambiente com 6400 msg/s verificou-se gargalos para todas as quantidades de *threads* testadas.

Como trabalhos futuros, pretende-se realizar comparações entre o Apache Camel e outras ferramentas de integração, como Spring Integration, Mule ESB, e Guaraná. Também estruturar novas experimentações com diferentes taxas de entrada, quantidades de threads, e sistemas operacionais. Estuda-se também a implementar um novo ambiente, onde seja possível enviar as mensagens de diferentes computadores.

Referências

- Frantz, R. Z., Corchuelo, R., and Arjona, J. L. (2011). An efficient orchestration engine for the cloud. In *2011 Third IEEE International Conference on Cloud Computing Technology and Science*, pages 711–716. IEEE.
- Frantz, R. Z., Corchuelo, R., and Roos-Frantz, F. (2015). A methodology to evaluate the maintainability of enterprise application integration frameworks. *International Journal of Web Engineering and Technology*, 10(4):334–354.
- Hohpe, G. (2005). Your coffee shop doesn't use two-phase commit [asynchronous messaging architecture]. *IEEE software*, 22(2):64–66.
- Hohpe, G. and Woolf, B. (2004). *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional.
- Messerschmitt, D. G., Szyperski, C., et al. (2005). Software ecosystem: understanding an indispensable technology and industry. *MIT Press Books*, 1.
- Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1):7–18.