SALÃO DO CONHECIMENTO UNIJUÍ 2018
CIÊNCIA PARA A REDUÇÃO DAS DESIGUALDADES

XXVI Seminário de Iniciação Científica
XXIII Jornada de Pesquisa
XIX Jornada de Extensão
VIII Mostra de Iniciação Científica Júnior
VIII Seminário de Inovação e Tecnologia

01 a 04 de outubro de 2018

Evento: XXIII Jornada de Pesquisa

FUNÇÃO OBJETIVO PARA OTIMIZAR O PROCESSAMENTO DE UMA MENSAGEM EM PROCESSOS DE INTEGRAÇÃO DE APLICAÇÕES[1]
OBJECTIVE FUNCTION TO OPTIMIZE THE PROCESSING OF A MESSAGE IN APPLICATION INTEGRATION PROCESSES

Francisco Rodrigues Lima[2], Rafael Z. Frantz[3]

[1] Pesquisa desenvolvida no curso de Mestrado em Modelagem Matemática da Unijuí
[2] Mestrando do Programa de Pós-Graduação em Modelagem Matemática
[3] Professor do Programa de Pós-Graduação em Modelagem Matemática

Resumo - As empresas, formam o ecossistemas de software, para oferecerem suporte aos processos de negócios. O desafio da Integrações de Aplicações Empresariais - EAI, é fazer com que diferentes aplicações que não foram desenvolvidas com o objetivo de integração, possam colaborar umas com as outras. As plataformas de integração implementam processos de integração, que fazem com que as aplicações funcionem de forma sincroni- zada, oferendo acesso às informações e funcionalidades de forma rápida e segura. As plataformas de integração geralmente fornecem uma linguagem de domínio específico, um kit de ferramentas de desenvolvimento, um motor de execução e uma ferramenta de monitoramento. Devido à heterogeneidade do ecossistema de software, há uma constante preocupação com a melhoria das plataformas de integração, a fim de que elas proporcionem a combinação de menor tempo de execução das tarefas de um processo de integração com a economia do consumo de recursos computacionais. O desempenho das plataformas EAI, dependem da performance dos motores de execução, que é um componente de software, responsável pela execução do processo de integração. O tempo de execução resulta da eficiência do motor. A eficiência do motor em escalar recursos e executar tarefas de integração tem um impacto direto no desempenho do processo de integração, e este é um dos desafios enfrentados pelas plataformas de integração. Nossa revisão literária identificou os motores de integração adotam algoritmos de agendamento de tarefas baseados em heurísticas, como a de prioridade e a First-In- First-Out (FIFO). Este trabalho propõe um modelo matemático, que resulta numa função objetivo, observando o tempo total de execução de um processo de integração, e que este tempo possa ser minimizado. Visto que nos atuais cenários, a exigência de qualidade em prestação de serviços das plataformas de integrações, remete ao tempo de processamento de mensagens pelo motor de execução.

Palavras-chaves: Enterprise Application Integration; Motor de Execução; Processa- mento de mensagem.

Evento: XXIII Jornada de Pesquisa

Abstract - Companies form the software ecosystem to support business processes. The challenge of Enterprise Application Integration - EAI, is to make different applications that were not developed with the purpose of integration, to collaborate with each other. Integration platforms implement integration processes, which make the applications work in a synchronized way, offering access to information and functionalities quickly and securely. Integration platforms typically provide a specific domain language, development toolkit, runtime engine, and monitoring tool. Due to the heterogeneity of the software ecosystem, there is a constant concern with the improvement of the integration platforms, so that they provide the combination of less time of execution of the tasks of an integration process with the economy of the consumption of computational resources. The performance of the EAI platforms depends on the performance of the execution engines, which is a software component, responsible for the execution of the integration process. The runtime results from the efficiency of the motor. The efficiency of the engine in scaling resources and performing integration tasks has a direct impact on the performance of the integration process, and this is one of the challenges faced by integration platforms. Our literary review has identified the integration engines adopt task scheduling algorithms based on heuristics, such as Priority and First-In-First-Out (FIFO). This work proposes a mathematical model, which results in an objective function, observing the total execution time of an integration process, and that this time can be minimized. Since in the current scenarios, the requirement of quality in service delivery of integration platforms, refers to the time of message processing by the execution engine.

Keywords: Enterprise Application Integration; Engine of Execution; Message processing.

1 Introdução

Nas empresas, a utilização dos ecossistemas de software (MANIKAS, 2016), como suporte aos processos de negócios, é constante. Geralmente o ecossistema de software é composto por aplicações que não colaboram entre si. A área que viabiliza a integração das aplicações é denominada Integração de Aplicações Empresariais (EAI). Esta área proporciona metodologias, técnicas e ferramentas para construir soluções de integração . Um processo de integração conecta várias aplicações num fluxo de integração composto por distintas tarefas, que processam dados encapsulados em mensagem. Uma tarefa executa operação atômica que implementam os padrões de integração (HOHPE; WOOLF, 2004), e estão dessincronizadas através de canais comunicação seguindo uma arquitetura Pipes-and- Filters (DOOLEY, 2011).

A construção de um processo de integração é realizada através das plataformas de integração,

como a Guaraná SDK (FRANTZ; CORCHUELO, 2012), Apache Camel (IBSEN; ANSTEY, 2010), WSO2 ESB (INDRASIRI, 2016), o Spring Integration (WALLS;

BREIDENBACH, 2005), Fuse (MYROSHNYCHENKO et al., 2013) e Mule (DOSSOT; D'EMIC; ROMERO, 2014). Tais plataformas realizam prodecimentos para projetar e executar soluções de integração. As plataformas de integração disponibilizam as seguintes ferramentas de construção: linguagem de domínio específico (DSL), utilizada para criar o modelo conceitual de uma processo de integração; um kit de desenvolvimento de soft- ware (SDK); motor de execução, responsável por executar um processo de integração, e ferramentas de monitoramento, que registra os valores estatísticos de desempenho e log de erros.

Os motores de execução, orquestram o processamento das mensagens nas tarefas. Esses motores tem a disposição recursos computacionais, a saber, pools de threads, com diferentes números de threads, onde uma thread é a unidade básica de processamento. Os motores de execução, podem ser baseados em dois modelos para orquestrar o processamento de mensagens; o modelo de pool global de threads, e o modelo de pools locais de threads. No modelo de pools locais de threads, é estabelecido que cada tarefa estará associada a um pools locais de threads, e que as mesmas possuem ainda, seus diferentes tempos de execução. O tempo de execução da tarefa, esta relacionado ao tipo trabalho a ser realizado, ou seja, em um fluxo, as tarefas poderão realizar trabalho individual de enrique, replicar ou filtrar uma mensagem, e poderá acontecer que o tempo de execução de uma tarefa seja maior que sua antecessora. No processo de integração, as tarefas são independentes, ligadas por canais, denominados slots. Os slots são responsáveis pela transferência de mensagens entre as tarefas.

Mediante a quantidade de mensagem que entram no processo de integração, onde uma tarefa 2**,** tem o tempo de execução maior que a antecessora tarefa 1, tal demora resultara no acmulo de mensagens esperando para serem processadas. E isto implicará no tempo total de execução de todo o processo de integração, tornando-o lento. No cenário em que o motor de execução, esta baseados no modelo de pools locais de threads para processar mensagens, e considerando que o pool local, possui um número de threads, espera-se que o motor escalone mais threads, para que paralelo possam consumir mensagens nos slots das tarefas. Contudo relacionar o número ideal de threads no pool local, com o tempo de execução em tarefas, não é algo trivial, requer uma distribuição ótima de threads pools locais de threads, e função objetivo que minimize o tempo total de execução do processo de integração.

Neste trabalho propõe-se descrever, o tempo total de execução do processo de integração, propor a função objetivo. Este artigo está organizado da seguinte maneira, a Seção 2 apresenta a formulação do problema, na seção 3 o modelo matemático da função objetivo, e por fim na seção 4, a conclusão.

2 Descrição do Problema

No processo de integração, arquitetado no estilo, Pipes-and-Filters (DOOLEY, 2011), pipes são representados por canais de mensagens e os filters por tarefas atômicas que implementam um padrão de integração concreto e processa dados encapsulados em mensagens. As mensagens entram no fluxo do processo de integração por porta de entrada e chegam nas tarefas por canais de mensagens. Para a mensagem ser processada por tarefas o motor de execução requisita threads que estão armazenadas em pools . Figura 1 representa esse estilo arquitetural.

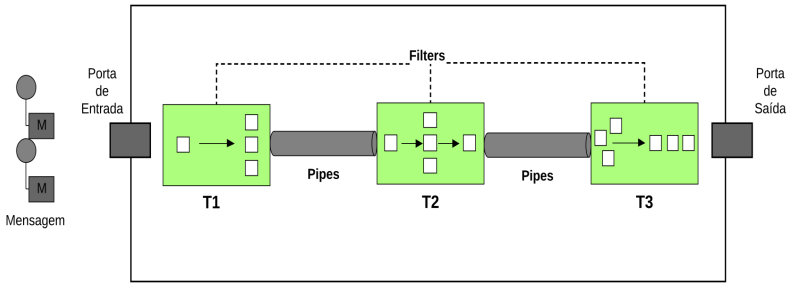

Figura 1. Estilo arquitetural *Pipes&Filters*

A maioria dos motores das de execução de plataformas open-source estão implementados com a tecnologia Java (Gosling, 2000) , que proporciona uma Application Programming Interface (API) denominada Executor para criação e alocação de unidades de threads, que estão, armazenadas em pools de threads. O dimensionamento do tamanho dos pools é uma configuração que precisa ser realizada pelos engenheiros de software. Nisto os motores de execução podem ser baseados em dois modelos; o modelo de pool global de threads, e o modelo de pools local de threads. As configurações nas quais os motores estão baseado, diferem nas seguintes características; o modelo de pool global de threads, caracteriza-se basicamente por apresentar um único pool de threads para executar todas as tarefas do processo de integração, enquanto o modelo de pools locais de threads , é caracterizado por apresentar, pools de threads associado a cada tarefa presente no processo de integração. Neste trabalho, será considerado o modelo conceitual de de pools locais de threads. A Figura 2 representa Motor de execução, baseado em pools locais de threads.

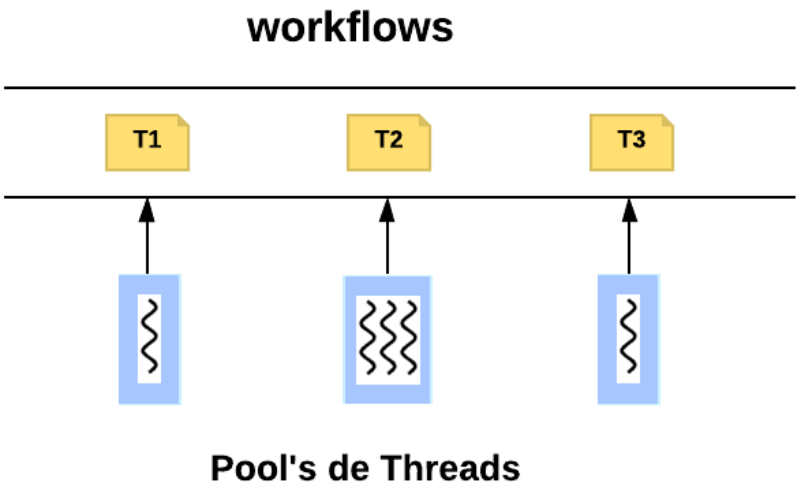

Figura 2. Motor de execução, configurado em *pools de threads* local.

Consideramos que a taxa de mensagens é, uma quantidade de mensagens que chegam nos slots das tarefas, por um intervalos de tempo. Quando taxa de mensagem é muito alta, o motor de execução, tende chamar as threads, para executar as tarefas dispostas no fluxo do processo de integração. Assumimos que cada tarefa, seja **Tarefa 1, Tarefa 2, Tarefa 3**, possua uma de unidade de tempo **U**t, e que essa unidade de tempo está relaciona com a complexidade do trabalhos por elas realizado, então nesse fluxo apresenta-se tarefas com unidade de tempo **Ut**, de valores diferentes. As tarefas com unidade de tempo **Ut**, relativamente pequena, realizará o processamento mais mensagens em menos tempo, enquanto que, as tarefas com unidade de tempo Ut, relativamente grande, demorariam muito para processar uma mensagem. O que pode acarretar no acúmulo ou, na predominância de mensagens nos slots das tarefas, a serem processadas, tornando o tempo de execução de toda o processo de integração, lento. Por exemplo uma mensagem M1, demoraria tempo considerável para ser processada do inicio ao fim, do fluxo. Portanto minimizar o tempo, em que uma mensagem espera no slot para ser processada, consistirá também em minimizar o total de execução do processo de integração. Assim, o problema pode ser formulado como: encontrar uma função objetivo do tempo total de execução, para otimiza o processamento de uma mensagem em processo de integração.

Pontanto desempenho dos motores de execução, poderá ser classificado com base no tempo total de execução de solução de integração. Visto que a distribuição de threads nos pools local, determina a variação do *makesspan*. Assim, o problema pode ser formulado como: encontrar uma função objetivo minimizando com o tempo total de execução da solução de integração, sem exceder um valor pré-definido para o total de threads distribuídas em cada pool local.

3 Modelo Matemático da Função Objetivo

Processar todas as tarefas em um fluxo de trabalho é sempre um trabalho complexo e crítico (Bilgaiyan, Sagnika, & Das, 2014). A quantidade de dados produzidos por aplicações de softwares, inclusos em uma soluções de integração, são relativamente grandes, de modo que, como conseqüência, o tempo médio de processamento, ou *Makespan,* será alto quando considerado que esses dados, esperam numa fila para serem anotados, depois processados, e ainda transferidos entre as tarefas.

O fluxo de trabalho, poderá ser representado na forma de gráfico acíclico direcionado $\boldsymbol{DAG = (Z, H)}$, onde $\boldsymbol{Z}$ é definido como o conjunto de tarefas $\mathbf{T = \{}\boldsymbol{T_1, T_2, ..., T_n}\mathbf{\}}$ e $\boldsymbol{d}$ é definido como arestas que representa dependência entre recursos. assumimos que um conjunto de recursos de processamento é representado por $\mathbf{R = \{}\boldsymbol{R_1, R_2,...., R_n}\mathbf{\}}$. Supondo que cada recurso de processamento $\boldsymbol{Ri}$ tenha seu próprio armazenamento, indicado por $\mathbf{DR = \{}\boldsymbol{DR1, DR2, DR3, DR4..., DRn}\mathbf{\}}$, e que todos os recursos de processamento estão localizados em P locais diferentes.

Em sequência o tempo total de execução, é a soma do tempo de execução de cada tarefa em cada recurso, e o tempo de transferência de dados entre uma tarefa pai e sua tarefa filha dependente, assumi-se que tais valores são hipoteticamente conhecidos (Bilgaiyan, Sagnika, & Das, 2014). A Figura 3 representa um fluxo de trabalho que consiste em 8 tarefas, onde $\boldsymbol{d}$ representa o tempo de transferência de dados $\boldsymbol{T_i}$, tarefa pai e $\boldsymbol{T_j}$, tarefa filha, com respetivos pool de threads local.

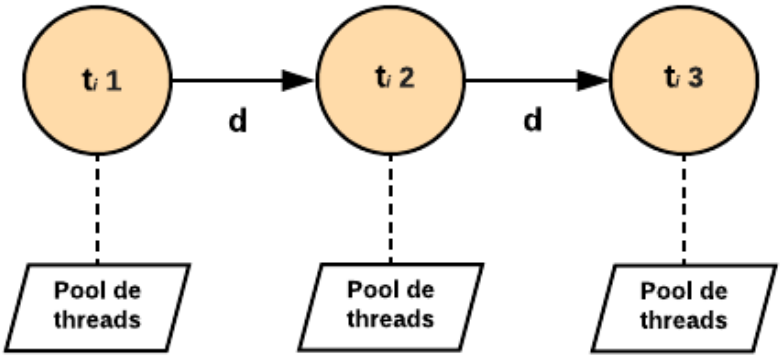

Figura 3: Representação do fluxo de trabalho.

Com objetivos básicos de agendamento de tarefas, o mapeamento tende fornecer acesso aos recursos disponíveis de forma que seja mantida a execução das tarefas com prioridade mais alta, considerando que há um conjunto limitado de recursos computacionais.

Então o tempo total do processamento de todo fluxo da solução $\boldsymbol{T_t (R_i)}$, é o somatório do tempo de espera na fila $\boldsymbol{T_e}$, *com* tempo incorrido da execução da tarefa $\boldsymbol{T_i}$ para o $\boldsymbol{R_i}$ de recursos, e o

tempo de tempo de transmissão T_m entre um conjunto de tarefas dependentes, calculado pela equação:

(1)

Assim, a função objetiva descrita abaixo, pode ser expressa, por minimizar o tempo total de execução T_t, da solução de integração, com complexidade da sua implementação para não ultrapassar o total de recursos disponíveis (threads), ou seja respeitar o parâmetro de restrição. A formulação é representada pela equação 2:

$$Minimize(T_t) \quad (2)$$

4 Conclusão

Neste artigo, referenciamos as configurações de processamento do motor de execução, pautado na disposição das threads, em pool global e pools local, buscando prover um modelo matemático que possibilite a minimização do tempo de execução da solução de integração. A abordagem foi formulada a partir da quantidade limitada de unidades de processamento threads, e a distribuição das mesmas, nos Pools local, em motores que implementa tecnologia JAVA. O artigo demostra ainda que a execução de tarefas da soluções de integração, bem como o escalonamento de recursos, consome tempo considerável.

O somatório das unidades de tempo, a saber, tempo incorrido na tarefa ***Ti***, quanto ao tempo de espera ***Te***, e ainda, o tempo de transmissão de dados ***Tm***, resulta numa função objetivo, o que aponta, que a minimização do tempo de execução resulta da combinação síncrona, entre os elementos contidos no motor de execução, obtendo assim uma economia considerável dos recursos computacionais.

Observou-se, que a compreensão do tempo total de processamento, descrito na função proposta, nos motiva a aprofundar o estudo, com a implementação de técnicas de otimização nos motores de execução das plataformas de integração.

5 Referências

BILGAIYAN, S.; SAGNIKA, S.; DAS, M. Workflow scheduling in cloud computing environment using cat swarm optimization. In: IEEE. *Advance Computing Conference*

(IACC), 2014 IEEE International. [S.l.], 2014. p. 680–685. Citado na página 5. DOOLEY, J. *Software development and professional practice*. [S.l.]: Apress, 2011. Citado 2 vezes nas páginas 2

and 3.
DOSSOT, D.; D'EMIC, J.; ROMERO, V. *Mule in action*. [S.l.]: Manning Publications Co., 2014. Citado na página 2.

FRANTZ, R. Z.; CORCHUELO, R. A software development kit to implement integration solutions. In: ACM. *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. [S.l.], 2012. p. 1647–1652. Citado na página 2.

HOHPE, G.; WOOLF, B. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. [S.l.]: Addison-Wesley Professional, 2004. Citado na página 2.

IBSEN, C.; ANSTEY, J. *Camel in action*. [S.l.]: Manning Publications Co., 2010. Citado na página 2.

INDRASIRI, K. *Beginning WSO2 ESB*. [S.l.]: Apress, 2016. Citado na página 2. MANIKAS, K. Revisiting software ecosystems research: A longitudinal literature study. *Journal of Systems and Software*, v. 117, p. 84 – 103, 2016. Citado na página 2.

MYROSHNYCHENKO, K. et al. *Maturidade de plataformas SOA Open Source vs proprietários*. Dissertação (Mestrado) — escola de comunicação, arquitetura artes e tecnologias da informação, 2013. Citado na página 2.

WALLS, C.; BREIDENBACH, R. *Spring in action*. [S.l.]: Dreamtech Press, 2005. Citado na página 2.