Uso de Anotações Semânticas para Exploração de Paralelismo em Workflows Intensivos em Dados*

Elaine Naomi Watanabe[1], **Kelly Rosa Braghetto**[1]

[1]Departamento de Ciência da Computação – Instituto de Matemática e Estatística
Universidade de São Paulo (USP)
Rua do Matão, 1010, Cidade Universitária, 05508-090 – São Paulo, SP, Brasil

{elainew,kellyrb}@ime.usp.br

Abstract. *Applications that analyze large volumes of data are often modeled as interconnected activities (workflows) and executed on high-performance platforms. Data partitioning and replication can make the activities parallelizable. However, to define a model that results in an efficient use of the platform is not trivial. This paper proposes semantic annotations to characterize the data processing in workflows activities, in order to automatically create strategies to parallelize the execution. In experiments with a workflow that handles 5.8 millions of data objects in a NoSQL system, the parallelism obtained from the annotations has reduced the makespan by 88.4% and the financial cost by 10.4%.*

Resumo. *Aplicações que analisam grandes volumes de dados costumam ser modeladas como atividades interligadas (workflows) e executadas em plataformas de alto desempenho. O particionamento e replicação de dados podem tornar as atividades paralelizáveis, mas definir um modelo que faça o uso eficiente da plataforma não é trivial. Este trabalho propõe anotações semânticas para caracterizar o tipo de processamento de dados feito em atividades de workflows e assim criar automaticamente estratégias para paralelizar a execução. Em experimentos com um workflow que manipula 5,8 milhões de objetos de dados em um sistema NoSQL, a paralelização obtida das anotações reduziu em 88,4% o tempo de execução do workflow e em 10,4% o custo monetário.*

1. Introdução

Na ciência moderna e na indústria, a necessidade da análise de grandes volumes de dados vem se tornando cada vez mais frequente. Processos de análise de dados geralmente são modelados como atividades interligadas por meio de fluxos de dados – os *workflows*. Workflows intensivos em dados têm alto custo computacional e, para melhorar a eficiência de suas execuções, diferentes abordagens já foram propostas, tais como o agrupamento de atividades [Singh et al. 2008] e a paralelização da execução [Pautasso and Alonso 2006].

A estrutura do workflow define o paralelismo da aplicação e, em geral, um SGWf – Sistema de Gerenciamento de Workflows – desconhece o tipo de processamento a ser realizado, por isso não é capaz de explorar estratégias para execução paralela automaticamente. As atividades paralelizáveis são definidas pelo usuário em tempo de projeto e criar uma estrutura que faça uso eficiente do ambiente de execução não é uma tarefa trivial.

*Esta pesquisa foi financiada pela CAPES e pelo NAPSoL-PRP-USP. Os autores agradecem também ao Google pelos créditos concedidos para uso de sua plataforma de nuvem.

Este trabalho propõe o uso de anotações semânticas para caracterizar as atividades de um workflow quanto ao tipo de processamento de dados que estas realizam e, assim, possibilitar a criação automática de estratégias que explorem o paralelismo de dados, considerando também informações sobre o ambiente de execução. O método proposto gera réplicas das atividades anotadas e define um esquema de indexação e distribuição dos dados do workflow que possibilita maior acesso paralelo.

Para avaliação do método, uma versão foi implementada para workflows executados no SGWf Pegasus 4.6.0. Foram realizados experimentos na plataforma de nuvem do Google, com um workflow que manipula 5,8 milhões de objetos de dados. Usou-se um SGBD relacional (PostgreSQL 9.3) e um NoSQL (MongoDB 3.2.4) em cenários com diferentes configurações de particionamento e replicação de dados. Os resultados obtidos indicam que a paralelização da execução das atividades promovida pelo método reduziu o tempo de execução do workflow em até 88,4% e seu custo monetário em até 10,4%.

2. Método para Exploração do Paralelismo de Dados em Workflows

O método proposto neste trabalho recebe como entrada um workflow orientado a dados descrito na forma de um Grafo Acíclico Dirigido (DAG), cujos vértices representam as atividades do workflow e os arcos definem o fluxo de dados entre essas atividades. Cada atividade pode receber como entrada e gerar como saída uma *lista de objetos de dados* armazenados em um sistema gerenciador de bancos de dados (SGBD).

As atividades podem ter anotações que as caracterizam quanto ao tipo de processamento realizado e aos atributos de dados que acessam. A partir das anotações sobre os tipos de processamento, realiza-se uma reestruturação do workflow, adicionando nele réplicas de atividades e configurando-as para receber subconjuntos dos seus dados de entrada originais. Depois, as anotações sobre atributos são usadas para a definição de índices que auxiliem as consultas e de um esquema de distribuição de dados apropriado para o workflow em questão. Essas etapas são descritas nas seções a seguir.

2.1. Caracterização de Atividades por Meio de Anotações Semânticas

Para o caracterizar o tipo de processamento das atividades, as anotações são:

- *Processamento por objeto – PO:* indica que a atividade processa um objeto de dados do seu conjunto de objetos de entrada individualmente.
- *Processamento por grupo de objetos – PG($\mathcal{L}$):* indica que a atividade processa os objetos de entrada em grupos definidos pelo(s) atributo(s) agrupador(es) listados em $\mathcal{L}$.

As anotações para os atributos dos objetos são referidas como:

- *Seleção de atributos – SA($\mathcal{L}$):* indica, em $\mathcal{L}$, os nomes dos atributos dos objetos de entrada que serão processados por cada atividade.
- *Ordenação de objetos – OO($\mathcal{L}$):* informa, em $\mathcal{L}$, os nomes dos atributos que serão utilizados para a ordenação dos objetos de dados de entrada da atividade.

O usuário define também anotações gerais sobre o workflow, como informações sobre a conexão com o banco de dados, o nome da coleção que contém os objetos de dados de entrada (*coleção de entrada*), a coleção que irá receber os objetos de dados resultantes do processamento (*coleção de saída*) e o número de nós de execução.

2.2. Reestruturação do Workflow Baseada nas Anotações

Uma atividade A com a anotação *PO* permite que o método crie n réplicas dessa atividade. O número n de réplicas equivale ao total de nós de execução disponíveis para o workflow. O método define ainda o subconjunto de objetos de dados que cada atividade irá processar e fornece essa informação como um parâmetro definido em um arquivo de configuração.

A anotação *PG($\mathcal{L}$)* sobre uma atividade possibilita a criação de tantas réplicas dessa atividade quanto for o mínimo entre o total de grupos e o número de nós de execução. O método associa um ou mais grupos a cada réplica pelo arquivo de configuração.

A anotação OO($\mathcal{L}$) permite que o método proposto crie índices para as coleções de objetos de dados que serão utilizadas como entrada em cada atividade, permitindo que os objetos sejam devolvidos às atividades na ordem definida pelo usuário nessa anotação.

3. Avaliação do Método Proposto

Para a avaliação do método proposto neste trabalho, foram analisados o tempo total de execução (*makespan*) e o custo monetário da execução de um workflow em 11 cenários diferentes, usando um banco de dados relacional e um orientado a documentos (NoSQL).

3.1. Workflow Utilizado

O workflow usado nos experimentos realiza a análise de *logs* de um *cluster* do Google. Foram selecionados 2.892.970 registros da coleção $task\_events$, com informações sobre o tipo de evento e o total requerido de CPU e memória. O total de objetos gerados e manipulados em cada cenário avaliado foi de 5.785.953 objetos. A Figura 1(a) mostra o modelo do workflow criado para sumarização dos dados. As atividades B, C, D e G geram como saída objetos de dados cujos valores dependem de todos os dados na coleção de entrada, por isso, as anotações *PO* e *PG* não são aplicáveis. Na atividade *E*, a anotação *PO* indica que os objetos de dados são processados individualmente e a anotação *PG* define que *F* processa grupos de objetos de dados.

A estrutura do workflow gerada automaticamente a partir do processamento das anotações no modelo é mostrada na Figura 1(b). O número de réplicas criadas leva em conta o número n de nós disponíveis para a execução das atividades. As atividades *E_i*, com $1 \leq i \leq n$ representam as réplicas da atividade *E* original. As atividades *F_j*, com $1 \leq j \leq m$ são réplicas da atividade *F*, sendo m o menor valor entre n e o número de grupos processados por *F*. Cada atividade *E_i* (*F_j*) é configurada para processar uma partição diferente da coleção de dados de entrada de *E* (*F*), como descrito na Seção 2.2.

3.2. Cenários Avaliados

A Tabela 1 descreve os 11 cenários avaliados. Para definir o número de nós de execução em um cenário, considerou-se o grau de paralelismo no modelo original do workflow (que é 3, por causa das atividades B, C e D) e o número de partições no banco de dados. Em cenários com mais de um nó de execução, criou-se 3 nós de execução para cada partição do banco, a fim de gerar concorrência no acesso e possibilitar paralelismo no tratamento das requisições ao banco. Cada cenário gerou uma instância de workflow diferente, com configurações e estrutura específicas. Todas as instâncias foram executadas 5 vezes.

Para o uso do MongoDB distribuído (cenários de W-7 a W-11), foram alocadas 3 máquinas para manter o servidor de configuração (*configservers*) e mais 3 máquinas para

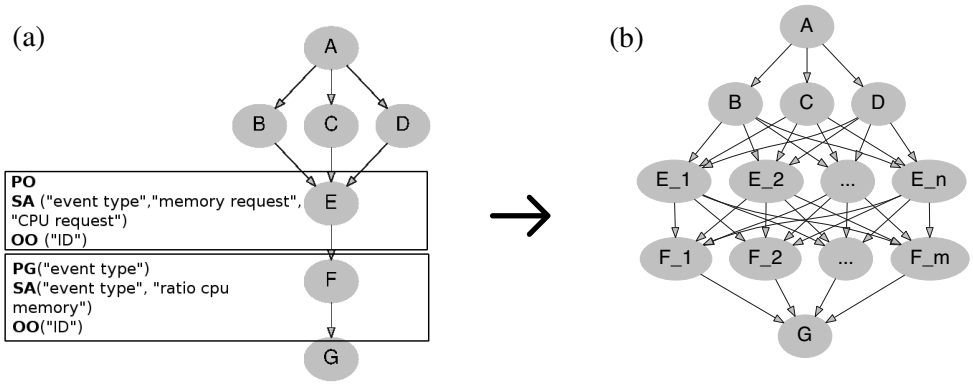

Figura 1. Modelo de workflow inicial (a) e o novo modelo (b) obtido por meio das anotações semânticas e de informações sobre o ambiente de execução.

Tabela 1. Caracterização dos cenários de execução avaliados.

	SGBD	Par-tições	Ré-plicas	Máq. SGBD	Nós Exec.
W-01	Postgres	1	1	1	1
W-02	Postgres	1	1	1	3
W-03[a]	Postgres	1	1	1	3
W-04	MongoDB	1	1	1	1
W-05	MongoDB	1	1	1	3
W-06[a]	MongoDB	1	1	1	3
W-07	MongoDB	1	3	9	3
W-08[a]	MongoDB	1	3	9	3
W-09	MongoDB	3	3	15	9
W-10[a]	MongoDB	3	3	15	9
W-11[a]	MongoDB	3	3	15	9

[a]Cenário envolvendo o uso de anotações semânticas no workflow.

os processos de roteamento das consultas com as partições (*mongos*). Além disso, foram usadas 3 máquinas de réplica para cada partição do banco de dados (mínimo recomendado): 1 servidor mestre e 2 escravos. O nível de consistência para escrita no MongoDB foi o padrão (confirmada a escrita na réplica mestre, a operação é considerada concluída). A leitura foi realizada somente em réplicas mestres, garantindo a consistência dos dados.

O nó central e os nós de execução do Pegasus usaram VMs *n1-standard-1*, com 20GB e 10GB de disco, respectivamente. Nos banco de dados centralizados, usou-se uma VM *n1-standard-1* com 50GB de disco para o SGBD. No MongoDB distribuído, para os *mongos* e *configservers* usou-se VMs *n1-standard-1*, com 10GB cada. Para os servidores *mongod*, usou-se máquinas *g1-small*, mais viáveis de serem escaladas (horizontalmente).

É necessário diferenciar os cenários W-10 e W-11, que possuem a mesma configuração de hardware. Em W-10, os objetos de dados manipulados são distribuídos por *hash* sobre seu atributo ID. Em W-11, as atividades anotadas como *PG* esperam que seu conjunto de entrada esteja distribuído por intervalo, conforme o(s) atributo(s) agrupador(es).

4. Resultados e Discussão

Foram comparados o tempo total de execução na Figura 2(a) e o custo monetário da execução do workflow na Figura 2(b) nos cenários descritos na Tabela 1. Pares de cenários selecionados são destacados na Tabela 2, indicando a diminuição ou aumento no *makespan* médio e no custo monetário médio do primeiro cenário em relação ao segundo.

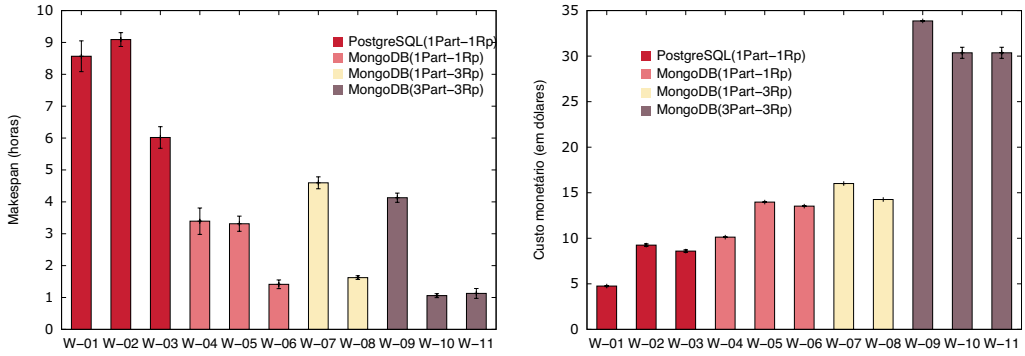

Figura 2. *Makespan* (a) e custo (b) médio da execução do workflow nos cenários.

Tabela 2. Comparação do desempenho em pares de cenários selecionados.

Cenários	*Makespan* Médio	Custo Médio
W-02 × W-01	+6,11%	+94,70%
W-03[a] × W-02	-33,77%	-7,13%
W-05 × W-02	-63,55%	+50,99%
W-05 × W-04	-2,36%	+37,91%
W-06[a] × W-02	-84,45%	+46,24%
W-06[a] × W-05	-57,34%	-3,15%

Cenários	*Makespan* Médio	Custo Médio
W-07 × W-05	+38,73%	+14,64%
W-08[a] × W-06[a]	+14,86%	+5,40%
W-10[a] × W-02	-88,37%	+228,16%
W-10[a] × W-09	-74,40%	-10,35%
W-11[a] × W-09	-72,69%	-10,35%

[a]Cenário envolvendo o uso de anotações semânticas no workflow.

As anotações propostas reduziram 33,77% do *makespan* e 7,13% do custo monetário no cenário com PostgreSQL (W-03 × W-02) e 57,34% do *makespan* e 3,15% do custo no cenário com MongoDB centralizado (W-06 × W-05). Além disso, com uma mesma quantidade de VMs com mesma configuração de hardware, obteve-se uma diminuição de 84,45% no *makespan* do workflow devido às anotações e ao uso do MongoDB, em substituição ao PostgreSQL (W-06 × W-02). Com a adição de máquinas – para se ter um banco de dados particionado e mais nós de execução no Pegasus – houve um ganho de desempenho ainda maior (redução de 88,36% no *makespan*, em W-10 × W-02).

A adição de réplicas dos dados no MongoDB gerou um aumento de 14,64% no custo monetário e de 38,73% no *makespan* (cenários W-07 × W-05), devido à sincronização entre as réplicas. Contudo, com as anotações, o aumento no *makespan* e no custo monetário médios foi menor – 14,86% e 5,40%, respectivamente (cenários W-08 × W-06), reduzindo o impacto negativo das réplicas. O custo monetário dos cenários com o MongoDB foi superior aos que usavam o PostgreSQL devido a um uso maior de discos.

Considerando os cenários em que existem mais de uma partição de dados (W-09, W-10 e W-11), observou-se uma redução no custo monetário de 10,35% tanto em W-10 × W-09 quanto em W-11 × W-09, devido ao cálculo do custo por hora. Entretanto, a redução no *makespan* em W-10 × W-09 (74,40%) é maior do que em W-11 × W-09 (72,69%). A distribuição por *hash* (cenário W-10) proporcionou um balanceamento de carga melhor entre as partições de dados do que a por intervalo (cenário W-11).

5. Breve Discussão sobre Trabalhos Relacionados

O trabalho de [Ferreira et al. 2014] compara os SGBDs PostgreSQL (relacional) e Cassandra (NoSQL) no contexto de proveniência em workflows científicos e destaca o bom desempenho do Cassandra em cenários distribuídos. Contudo, não aborda como os dados de entrada e saída das atividades são mantidos ou distribuídos nos SGBDs.

O artigo de [Dean and Ghemawat 2010] descreve o *MapReduce* – um modelo de programação para processamento de grandes volumes de dados. Diversos arcabouços implementam-no, provendo alta escalabilidade e interfaces para distribuição do processamento. No entanto, é um modelo para programadores e exige que o usuário domine uma linguagem de programação específica para a descrição do processamento a ser realizado.

No trabalho de [Ogasawara et al. 2011], é proposta uma abordagem algébrica para a descrição das dependências entre as atividades em um workflow. As atividades são representadas por operadores conforme sua forma de consumo e geração de dados; esses descrevem o modelo de workflow a ser utilizado por um escalonador, cujo objetivo é melhorar o desempenho da execução do workflow por meio de paralelização de atividades.

Os trabalhos aqui discutidos usam estratégias de escalonamento e modelos de programação distribuída para paralelizar o processamento ou avaliam o impacto do armazenamento da proveniência de dados. Na proposta deste artigo, o paralelismo dos dados é promovido por meio de modificações na estrutura do banco de dados e do workflow.

6. Considerações Finais

Este trabalho apresentou um método que combina anotações semânticas e informações do ambiente de execução para aumentar, de forma automática, o paralelismo no acesso aos dados na execução de workflows. A aplicação do método sobre um workflow que manipula 5,8 milhões de objetos de dados mantidos em um sistema NoSQL gerou uma redução de até 88,4% no tempo de execução do workflow. Além disso, considerando cenários com as mesmas configurações e número de máquinas virtuais, obteve-se uma redução do custo monetário da execução de até 10,4% e do makespan de até 74,4%.

Diferentemente de outras abordagens para a paralelização de workflows, o uso das anotações não depende de conhecimento específico em programação paralela. O próprio projetista do workflow, conhecedor do domínio modelado, é capaz de associar as anotações às atividades e, assim, obter um melhor desempenho da plataforma de computação.

Referências

Dean, J. and Ghemawat, S. (2010). MapReduce: a flexible data processing tool. In Communications of the ACM, volume 53, pages 72–77. ACM.

Ferreira, G. R. et al. (2014). Uso de SGBDs NoSQL na gerência da proveniência distribuída em workflows científicos. In The 29th Brazilian Symposium on Databases.

Ogasawara, E. et al. (2011). An algebraic approach for data-centric scientific workflows. In The VLDB Endowment, volume 4, pages 1328–1339.

Pautasso, C. and Alonso, G. (2006). Parallel computing patterns for grid workflows. In The 6th Workshop on Workflows in Support of Large-Scale Science, pages 1–10.

Singh, G. et al. (2008). Workflow task clustering for best effort systems with pegasus. In The 15th ACM Mardi Gras Conference, pages 9:1–9:8.