Proveniência de dados em workflow de Bioinformática com PROV-DM e armazenamento em banco de dados baseado em grafo

Rodrigo Pinheiro

Universidade de Brasília

Exame de Qualificação de Mestrado Programa de Pós-Graduação em Informática
Orientador(a): Maristela Terto de Holanda

16 de abril de 2014

Agenda

Workflow na Bioinformática

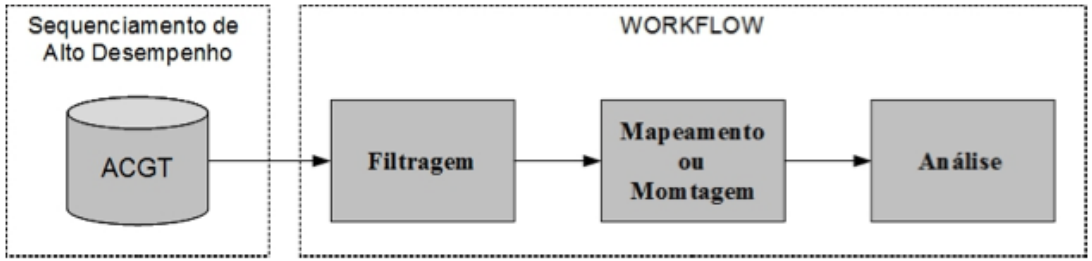

Figura: Exemplo de workflow para projetos genoma e transcritoma.

- A execução de um *workflow* pode levar dias ou semanas;
- É necessário re-executar um experimento e validá-lo;

O que é proveniência de dados?

Definição

O termo Proveniência de dados diz respeito à origem ou procedência dos dados. A proveniência também pode se referir à auditoria, triagem, linhagem e origem do dado [Buneman et al., 2001].

- Segundo [Davidson and Freire, 2008], pode-se dividir o nível em que é feita a captura conforme segue:
  - *Workflow*: envolve a descrição da execução de um processo;
  - Atividade: pode ocorrer de duas formas. Na primeira, cada processo executado é alterado para capturar os dados de proveniência. Na segunda, podem ser criados programas específicos para monitorar a execução de um determinado processo e capturar os dados de proveniência;
  - Sistema Operacional: utiliza os dados fornecidos pelo próprio sistema operacional como insumo para a proveniência.

Modelos de Proveniência

Definição

Tem como principal objetivo fornecer uma estrutura para que os dados de proveniência possam ser armazenados e recuperados, mantendo seu significado e potencializando os seus benefícios.

- Modelo W7: objetiva descrever as propriedades de um objeto de caráter geral;
- *Provenance Vocabulary*: volta a sua atenção para o problema da proveniência de dados publicados na *web*;
- *Provenir Ontology*: foi desenvolvido para ser um modelo de proveniência de dados genérico;
- OPM (*Open Provenance Model*): procura demonstrar a relação causal entre eventos que afetam objetos (digitais ou não) e descreve essa relação através de um grafo acíclico direcionado.

PROV-DM

- Teve a sua primeira versão desenvolvida em outubro de 2011, tornando-se uma recomendação do W3C em Abril de 2013;
- Tem como principal função descrever as pessoas, entidades e atividades envolvidas na produção de uma peça de dado ou de um objeto qualquer;
- Cria as condições para que a proveniência seja demonstrada e trocada entre diferentes sistemas;
- Demonstra a proveniência de qualquer objeto (real ou imaginário) através de um grafo direcionado;
- Símbolo para representar grandes conjuntos de dados;

Componentes

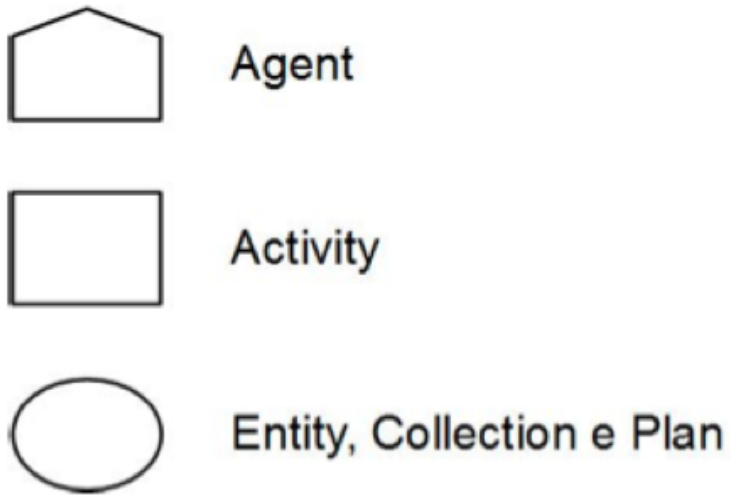

Figura: Representação gráfica dos diferentes tipos no modelo PROV-DM. [W3C, 2014]

- Entidades: podem representar qualquer objeto (real ou imaginário);
- Atividades: é algo que ocorre ao longo de um período de tempo e atua sobre ou com entidades;

Componentes

- Agente: são entidades que influenciam, direta ou indiretamente, a execução das atividades;
- Coleções: são entidades que possuem membros, os quais são também entidades;
- Anotações: fornece mecanismos para inclusão de anotações para os elementos do modelo;
- Plano: representa um conjunto de ações ou passos que um Agente deve seguir para chegar a um determinado objetivo;
- Conta: representa um conjunto de informações (tipos e relações) que compõe um grafo de proveniência.

Relações

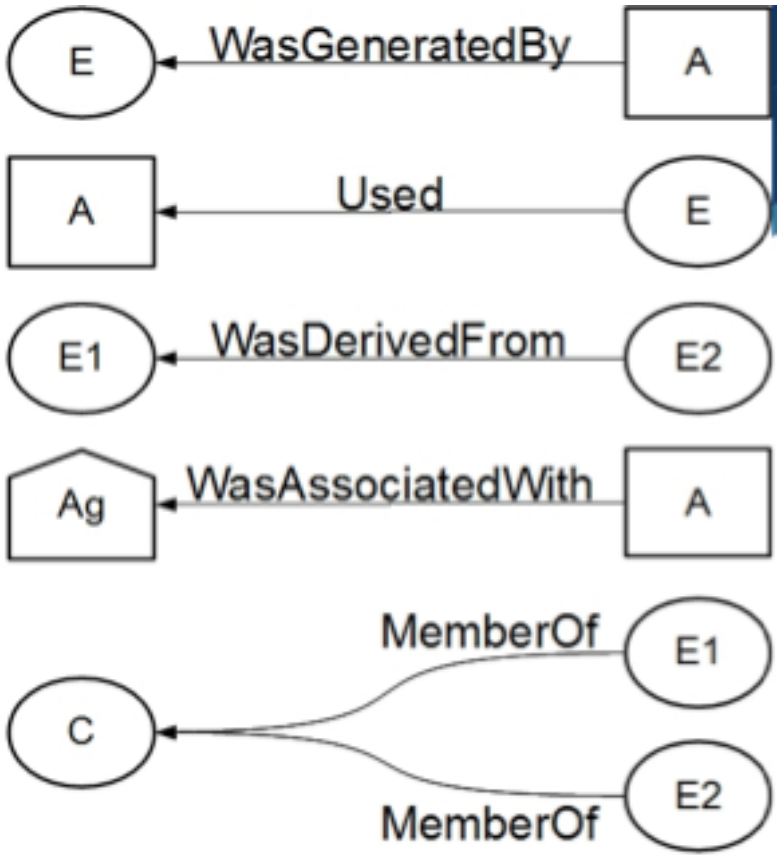

Figura: Representação gráfica das diferentes relações no modelo PROV-DM. [W3C, 2014]

Relações

- *used*: indica que uma Entidade foi usada por uma Atividade;
- *wasGeneratedBy*: indica que uma Entidade foi gerada por uma Atividade;
- *wasAttributedTo*: atribui algum tipo de responsabilidade a um Agente sobre uma Entidade;
- *wasAssociatedWith*: atribui algum tipo de responsabilidade a um Agente sobre uma Atividade;
- *wasDerivedFrom*: indica, de forma geral, que uma Entidade (original) foi usada, direta ou indiretamente, na geração de outra Entidade (derivada);
- *memberOf* : indica que uma determinada Entidade é membro de uma Coleção.

Restições

- Restrições de Atividade: restrições relacionadas à execução das Atividades.
  - Início/Fim: o início da execução de uma Atividade deve preceder o seu fim;
  - Uso: o uso de uma Entidade por uma Atividade deve ocorrer entre o início e o fim da sua execução;
  - Geração: a geração de uma Entidade por uma Atividade deve ocorrer entre o início e o fim da sua execução.
- Restrições de Agente - restrição relacionada ao ciclo de vida de um Agente.
  - Associação: a associação entre um Agente e uma Atividade deve ocorrer entre o início e o fim da execução desta Atividade.

Restições

- Restrições de Entidade - restrições relacionadas ao ciclo de vida de uma Entidade.
  - Geração/Uso: a geração de uma Entidade deve preceder o seu uso;
  - Derivação/Uso/Geração: para os casos em que existe uma derivação entre duas Entidades, por exemplo E2 é derivado de E1, e o uso de E1 é conhecido, então o uso de E1 deve preceder a geração de E2;
  - Derivação/Geração/Geração: para os casos em que existe uma derivação entre duas Entidades, por exemplo E2 é derivado de E1, e o uso de E1 não é conhecido, então a geração de E1 deve preceder a geração de E2.

Por que PROV-DM em projetos de Bioinformática?

De acordo com [de Paula et al., 2013] as razões de usar o modelo PROV-DM são:

- A aplicação do modelo PROV-DM em projetos de Bioinformática se mostrou simples e direta;
- Os componentes do modelo, tais como o agente, atividade e coleção, representam elementos presentes em grande parte dos experimentos executados em projetos de Bioinfomática;
- As relações, por sua vez, demonstram de forma objetiva as dependências entre cada elemento no grafo;
- Utilização das regras e do tipo de derivação permitem maior grau de especificidade quando necessário.

Ambiente de computação em nuvem

- O ambiente de computação em nuvem tem se tornado atraente para a execução de experimentos científicos devido:
  - Escabilidade;
  - Interoperabilidade;
  - Idéia de recursos infinitos.

Porém, informações como parâmetros de entrada e saída, tempo de execução, métodos invocados e processos iniciados e finalizados são importantes na execução de experimentos científicos, porque há a necessidade de validar o experimento e reproduzí-lo.

Bancos de dados no ambiente de computação em nuvem

Para um ambiente heterogêneo, distribuído e de alta disponibilidade como o de computação em nuvens, os bancos relacionais não apresentam um bom desempenho, surgindo os bancos de dados *NoSQL*.

- *NoSQL*
  - Gerenciam grades volumes de dados;
  - Em geral, fornecem garantias de consistência fraca, estruturas e interfaces simples.

Como um tipo de bancos de dados *NoSQL*, se destacam os bancos de dados de grafos que permitem o armazenamento de entidades e também relacionamentos entre essas entidades.

Classificação dos bancos *NoSQL*

- Os bancos *NoSQL* podem ser classificados em quatro classes gerais [Padhy et al., 2014]:
  - Chave/Valor: os dados são armazenados como pares chave-valor que são indexados para recuperação por chaves. Os mais populares são o *Riak*, *Redis*, *Memcached*, *Berkeley DB*, *Amazon DynamoDB*, *Porject Voldemort* e *HamsterDB*;
  - Orientado a coluna: armazena os dados em linhas com colunas associadas, fazendo uso de uma chave de linha. Exemplos são o *BigTable*, *Cassandra* e *HBase*;
  - Armazenamento baseado em documentos: os dados são armazenados e organizados como uma coleção de documentos. Exemplos são o *MongoDB*, *Apache CouchDB* e *RavenDB*;
  - Bancos de dados de grafos: permitem armazenar entidades e também relacionamentos entre essas entidades. Exemplos: *Neo4J*, *Infinite Graph*, *OrientDB* e *FlockDB*.

Modelo de dados de grafo

Definição

O modelo de dados de grafo é uma estrutura na qual o esquema e/ou instâncias são modeladas como um grafo dirigido, possivelmente rotulado, ou generalizações da estrutura de dados do grafo, onde a manipulação de dados é expressa por operações orientadadas para o grafo e construtores de tipos, e restrições de integridade que podem ser definidas sobre a estrutura do grafo. [Angles and Gutierrez, 2008]

Tem-se em [Angles and Gutierrez, 2008] as principais vantagens de se usar esse tipo de modelo:

- Permite uma modelagem mais natural dos dados porque as estruturas de grafos são visíveis para o usuário;
- Permite expressar as consultas em um nível maior de abstração;
- Permite a implementação de algoritmos eficientes para realizar específicas operações.

Componentes

- Entidade/Objeto: representa algo que existe como uma unidade simples e completa;
- Relação: é uma propriedade ou predicado que estabelece uma conexão entre duas ou mais entidades;
- Atributos: informações associadas a nós e arestas;
- Direcionamento: dependendo do problemas a relação entre dois nós pode ser simétrica ou não. Se a relação é simétrica, as duas pontas da aresta são diferentes, mas indistinguíveis, ou seja, não há ponta inicial nem ponta final. Se a relação não é simétrica, é possível diferenciar as duas pontas da aresta;
- Nós e arestas rotuladas: em algumas aplicações, é possível diferenciar rótulos (ou tipos) de nós e arestas.

Operações

- Travessias: são operações que se iniciam de um único nó e explora recursivamente os vizinhos até que uma condição seja alcançada, tais como a profundidade ou visitar um nó de destino;
- Análise Gráfica: basicamente inclui o estudo da topologia de grafos para analisar a sua complexidade e para caracterizar objetos do grafo;
- Transformação: compreende as operações que alteram o banco de dados de grafo. Cargas de um grafo, adicionar/remover nós ou arestas dos grafos, criar novos tipos de nós/arestas/ atributos ou modificar o valor de um atributo;
- Atributos: bancos de dados não só tem que gerir a informação estrutural do grafo, mas também dos dados associados às entidades do grafo;
- Resultado: grafos, agregados e conjuntos.

Banco de dados relacional versus Banco de dados de grafo

De acordo com [Vicknair et al., 2008]:

Tabela: Comparação MySQL versus Neo4J. Requisitos objetivos.

Banco de dados	Consultas estruturais	Buscas textuais	Consultas de contagem numérica
MySQL			x
Neo4J	x	x	

Tabela: Comparação MySQL versus Neo4J. Requisitos subjetivos.

Banco de dados	Maturidade	Flexibilidade	Suporte	Segurança
MySQL	x		x	x
Neo4J		x		

Banco de dados relacional versus Banco de dados de grafo

Em *Neo4J In Action*, [Partner et al., 2008] executou um experimento comparando um banco de dados relacional com o banco de dados de grafos Neo4J. Como resultado tem-se a tabela abaixo:

Tabela: Comparação MySQL versus Neo4J. [Partner et al., 2008]

Profundidade	Tempo de execução no MySQL (s)	Tempo de execução no Neo4J (s)	Registros retornados
2	0.016	0.01	2.500
3	30.267	0.168	110.000
4	1543.505	1.359	600.000
5	—	2.132	800.000

Contextualização da Proposta

Como apresentado em [de Paula et al., 2013], o modelo PROV-DM pode ser aplicado em workflow de Bioinformática onde através de um grafo é possível facilmente representar a proveniência em um experimento da Bioinformática salvando os dados em arquivos XML.

Em [Pinheiro et al., 2013] é proposto a captura automática de dados e o armazenamento em um esquema relacional baseado no modelo PROV-DM.

Como o PROV-DM é um modelo baseado em grafo, onde toda proveniência pode ser representada através de nós e arestas.

A execução do workflow em um ambiente de computação em nuvem é uma realidade em vários experimentos na Bioinformática.

Proposta

Realizar a proveniência de dados em projetos de Bioinformática utilizando um modelo de proveniência PROV-DM e armazenando os dados em bancos de dados de grafos no ambiente de computação em nuvem.

- Objetivos específicos
  - Definir uma arquitetura de proveniência de dados para um ambiente de computação em nuvem em projetos de Bioinformática, utilizando bancos de dados de grafos;
  - Implementar a arquitetura proposta;
  - Realizar estudo de caso com workflows científicos reais da Bioinformática;
  - Avaliar os resultados obtidos.

Fundamentação Teórica

- *Provenance for the cloud*. [Muniswamy-Reddy et al., 2010]: defende a proveniência como primeiro conjunto de dados a ser salvo no ambiente de computação em nuvem;
- Captura de Metadados de Proveniência para Workflows Científicos em Nuvens Computacionas. [Paulino et al., 2010]: apoia a coleta de metadados de proveniência em experimentos científicos para o ambiente de computação em nuvem;
- Reprodução de Experimentos Científicos Usando Nuvens. [de Oliveira et al., 2012]: visa a reprodução do ambiente onde o experimento computacional foi originalmente executado;
- *Capturing and querying workflow runtime provenance with prov: A practical approach.* [Costa et al., 2013]: propõe uma solução para monitorar o experimento durante a sua execução, usando dados de proveniência;

Fundamentação Teórica

- *Performance analysis of data filtering in scientific workflows.* [Goncalves et al., 2013]: propõe melhorar a performance de workflows científicos, reduzindo os dados a serem processados pelas atividades através de dados de proveniência;
- *Provenance in bioinformatics workflows.* [de Paula et al., 2013]: propõe uso de proveniência de dados com base no modelo PROV-DM para fluxos de trabalho de projetos genoma;
- *Achieving reproducibility by combining provenance with service and workflow versioning.* [Woodman et al., 2011]: trata de como um sistema de armazenamento de proveniência é usado pela plataforma de computação em nuvem e-Science Central;
- *Prob: A tool for tracking provenance and reproducibility of big data experiments.* [Korolev and Joshi, 2014]: usa uma ferramenta de captura de dados de proveniência para alcançar a reprodução de experimentos científicos envolvidos com *workflow* de *big data*, o PROB;

Resumo Referencial Teórico

Tabela: Resumo referencial teórico.

Proposta	**Modelo de proveniência**	**Ambiente de execução**	**Sistema de armazenamento**
[Muniswamy-Reddy et al, 2010]	—	Computação em nuvem	*Storage*
[Paulino et al., 2010]	OPM	Computação em nuvem	Banco de dados relacional
[de Oliveira et al., 2012]	PROV-DM	*Dekstop*	Banco de dados relacional
[Costa et al., 2013]	PROV-DM	*Dekstop* e Computação em nuvem	Banco de dados relacional
[Goncalves et al., 2013]	—	Computação em nuvem	Banco de dados relacional
[de Paula et al., 2013]	PROV-DM	*Dekstop*	Arquivo *XML*
[Woodman et al., 2011]	OPM	Computação em nuvem	Banco de dados de grafo
[Korolev and Joshi, 2014]	PROV-DM	Computação em nuvem	Sistema de versionamento *GitHub*

Arquitetura Proposta

- Existem na literatura trabalhos usando os modelos de proveniência OPM e PROV-DM;
- Armazenamento dos dados de proveniência em bancos de dados de grafos;
- Proveniência de dados para processos executados em um ambiente de computação em nuvem.

Porém não tratam de um modelo de dados para proveniência de workflow de Bioinformática usando bancos de dados de grafos para armazenar os dados do modelo PROV-DM com execução em um ambiente de computação em nuvem.

Modelo de dados em bancos de dados de grafo

- Os dois tipos básicos do modelo PROV-DM, atividade e entidade serão representados como nós no banco de dados de grafo, e serão diferenciados pela propriedade Tipo.
- Já as relações serão representadas pelas arestas, sendo diferenciadas também por uma propriedade Tipo.

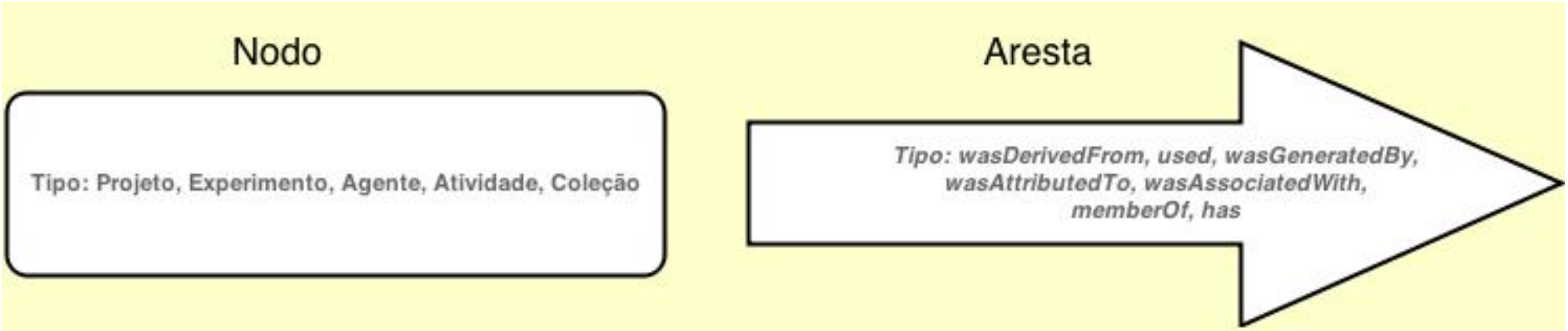

Figura: Mapeamento dos tipos e relações para o modelo de dados baseado em grafo.

Arquitetura Proposta

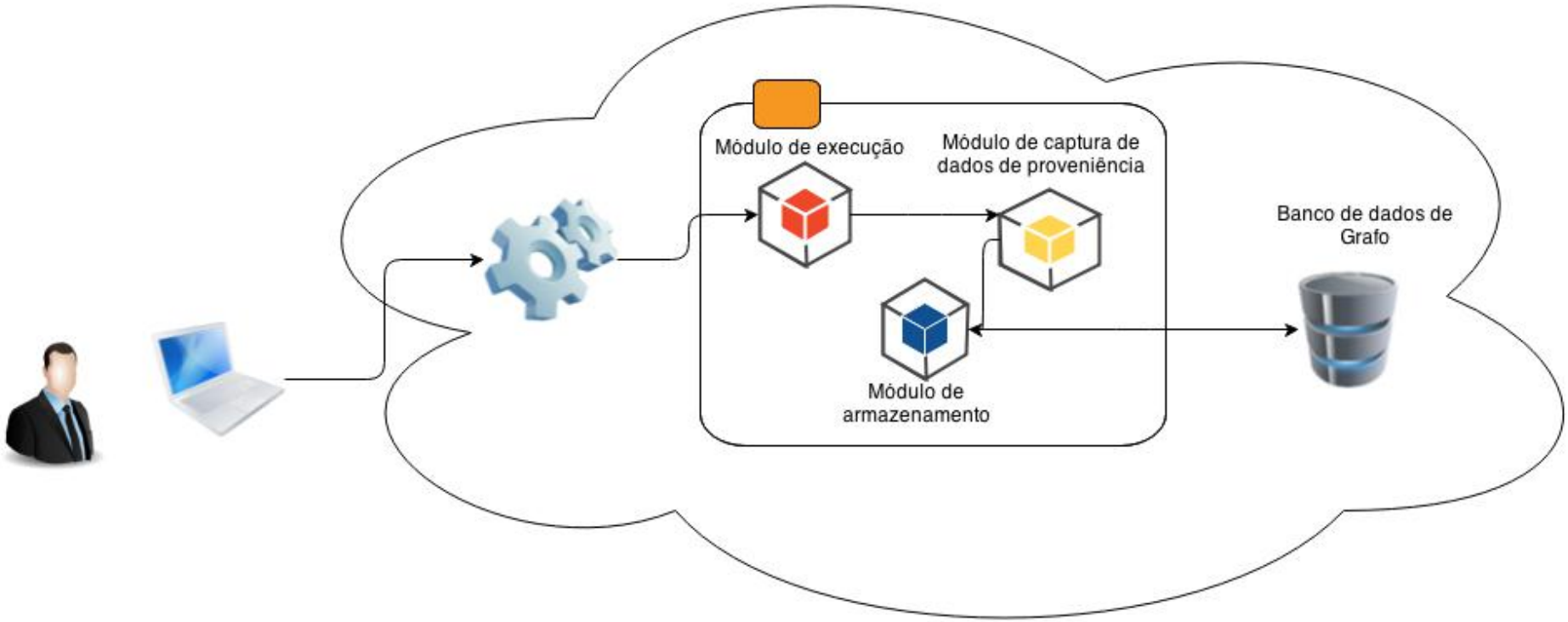

Figura: Arquitetura Proposta.

Arquitetura Proposta

- Uma interface web simples e amigável para o usuário, permitindo-o criar projetos, experimentos e atividades. Os dados informados pelo usuário serão enviados para a nuvem para serem processados;
- Módulo de execução: responsável por executar os comandos ou programas recebidos do usuário;
- Módulo de captura de dados de proveniência: responsável por receber os dados informados pelo usuário, interpretá-los e realizar a captura da proveniência de forma automática;
- Módulo de armazenamento: será responsável pela comunicação com o banco de dados de grafo e prover uma interface de acesso;
- Banco de dados de grafo: responsável pelo armazenamento dos dados na forma de grafo.

Metodologia

- Primeira fase: estudo e análise;
- Segunda fase: especificação da arquitetura;
- Terceira fase: implementação;
- Quarta fase: testes;
- Quinta fase: avaliação e correção;
- Sexta fase: publicação e defesa da dissertação.

Cronograma

Foi realizada a publicação do artigo [Pinheiro et al., 2013] na conferência *The IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* de 2013, que propõe a captura automática de dados de proveniência e o armazenamento em um esquema relacional baseado no modelo PROV-DM.

Tabela: Cronograma de atividades.

Etapa	**2013/1**	**2013/2**	**2014/1**	**2014/2**
1	X	X	X	
2		X	X	
3			X	
4			X	X
5				X
6				X

Referências

Buneman, P., Khanna, S., and Wang-Chiew, T. (2001).
Why and where: A characterization of data provenance.
Database TheoryICDT 2001, 316 – 330.

Davidson, Susan B and Freire, Juliana. (2008)
Provenance and scientific workflows: challenges and opportunities.
Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 1345–1350.

Tan, Wang Chiew. (2004)
Research Problems in Data Provenance.
IEEE Data Eng. Bull., 27(4), 45–52.

W3C. (2014)
www.w3.org/TR/prov-dm/
www.w3.org/TR/prov-dm/

Referências


Padhy, R. P., Patra, M. R., and Satapathy, S. C. (2011).
Rdbms to nosql: Reviewing some next-generation non-relational databases.
International Journal of Advanced Engineering and Technologies., 11(1), 015–030.


Angles, R. and Gutierrez, C. (2008).
Survey of graph database models.
ACM Computing Surveys (CSUR)., 40(1), 1.


Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., and Wil- kins, D. (2010).
A comparison of a graph database and a relational database: a data provenance perspective.
Proceedings of the 48th annual Southeast regional conference., 42.


Partner, J., Vukotic, A., and Watt, N. (2013).
Neo4j in Action.
OReilly Media..


de Paula, R., Holanda, M., Gomes, L. S., Lifschitz, S., and Walter, M. E. M. (2013).
Provenance in bioinformatics workflows.
BMC Bioinformatics. 14(Suppl 11):S6.

Referências

Muniswamy-Reddy, K.-K., Macko, P., and Seltzer, M. I. (2010).

Provenance for the cloud.

In FAST. 10, 15–14

Paulino, C., Oliveira, D., Cruz, S., Campos, M. L. M., and Mattoso, M. (2010).

Captura de metadados de proveniência para workflows científicos em nuvens computacionais.

Anais do XXV. Simposio Brasileiro de Banco de Dados. .

de Oliveira, A. H. M., de Souza Martins, M., Modesto, I., de Oliveira, D., and Mattoso, M. (2012).

Reprodução de experimentos científicos usando nuvens.

Anais do XVII Simposio Brasileiro de Banco de Dados (SBBD 2012)..

Costa, F., Silva, V., de Oliveira, D., Ocaña, K., Ogasawara, E., Dias, J., and Mattoso, M. (2013).

Capturing and querying workflow runtime provenance with prov: a practical approach.

Proceedings of the Joint EDBT/ICDT 2013 Workshops. 282–289

Referências

Goncalves, J., de Oliveira, D., Ocaña, K., Ogasawara, E., Dias, J., and Mattoso, M. (2013).

Performance analysis of data filtering in scientific workflows.

Journal of Information and Data Management. , 4(1):17.

Woodman, S., Hiden, H., Watson, P., and Missier, P. (2011).

Achieving reproducibility by combining provenance with service and workflow versioning.

Proceedings of the 6th workshop on Workflows in support of large-scale science. 127–136.

Korolev, V. and Joshi, A. (2014).

Prob: A tool for tracking provenance and reproducibility of big data experiments.

Reproduce14. HPCA 2014..

Pinheiro, R., Holanda, M., Araujo, F., Walter, E., and Lifschitz, S. (2013).

Automatic capture of provenance data in genome project workflows.

Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference. 15–20

Dúvidas?