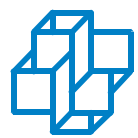
GB-500: Introdução a Workflows Científicos e suas Aplicações

Professores: Luiz Gadelha e Kary Ocaña

Programa de Pós-Graduação em Modelagem Computacional, P3/2015
Laboratório Nacional de Computação Científica

16 de junho de 2015



Laboratório
Nacional de
Computação
Científica

Experimentos Científicos Computacionais

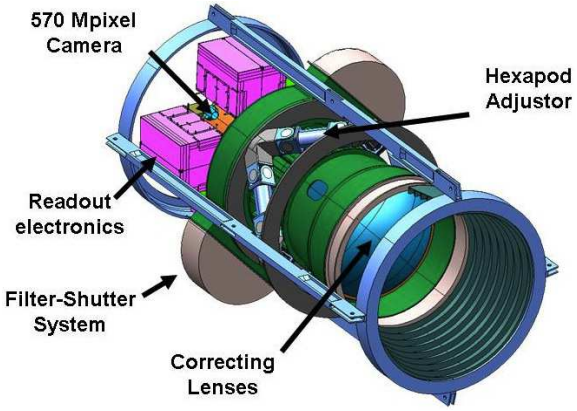
- Uma parte crescente da ciência e tecnologia se apoia em experimentos computacionais.
- Características destes experimentos:
  - grande quantidade de tarefas computacionais;
  - grande quantidade de dados;
  - dados armazenados em formatos diversos;
  - utilização de computação paralela e distribuída.

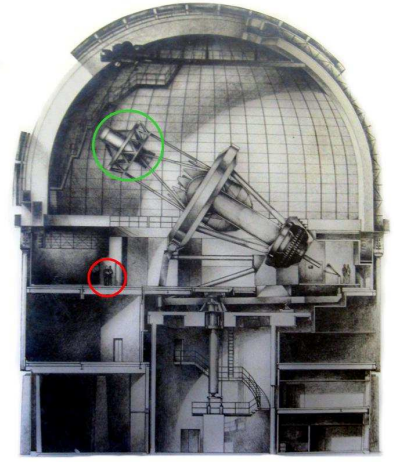
Introdução

- Exemplos:
  - Modelos climáticos mais recentes geram 8TB para uma simulação de 100 anos com resolução de 100km, e 8PB para resolução de 3km.
  - *Large Hadron Collider* gera 10PB de dados brutos por dia 15PB de dados tratados por ano.
  - O SKA (*Square Kilometre Array*) será composto por 3.000 rádio-telescópios de 15m de diâmetro. Envolve 70 instituições de 20 países. O volume de dados brutos gerado diariamente será de 1 Exabyte, (2x o que trafega pela Internet por dia atualmente). Volume pós-processamento: 300 Petabytes a 1,5 Exabyte por ano.

R. Kouzes et al. The Changing Paradigm of Data-Intensive Computing. *IEEE Computer*, 42(1):26-34, 2009.

Exemplo: Dark Energy Survey (DES)

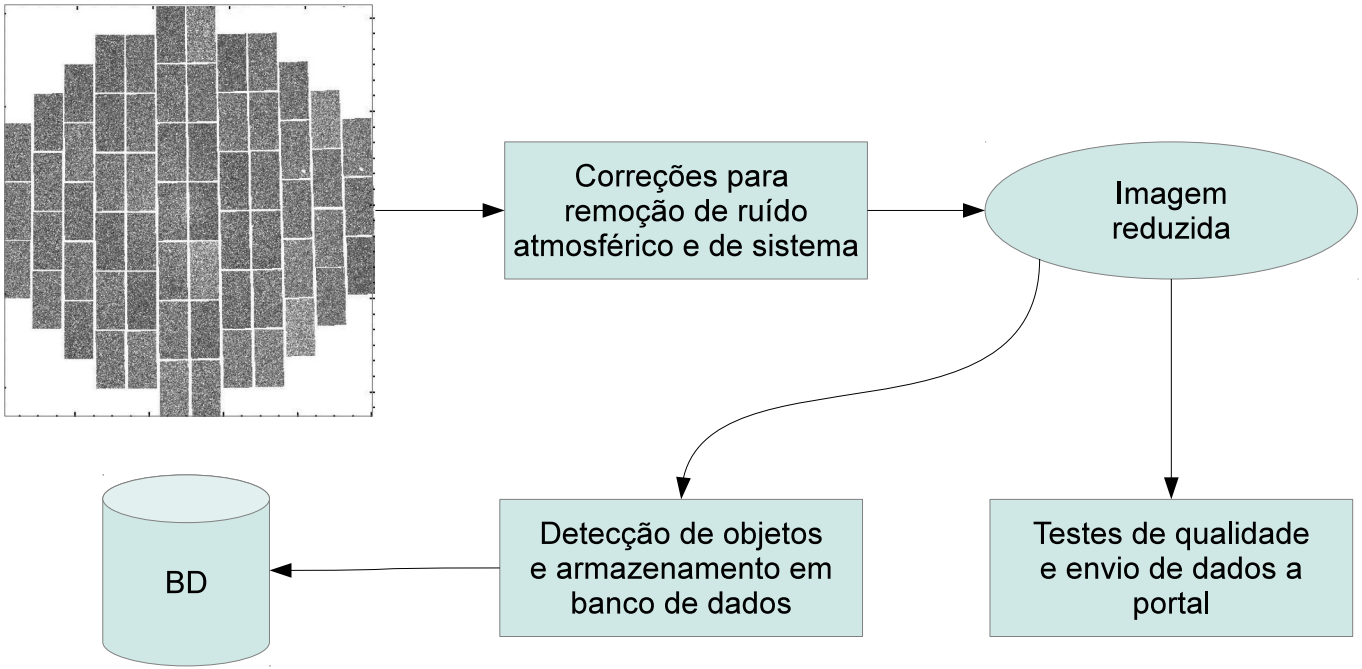




- Câmera de 570 megapixels.
- 400 imagens (6,6TB) por noite.

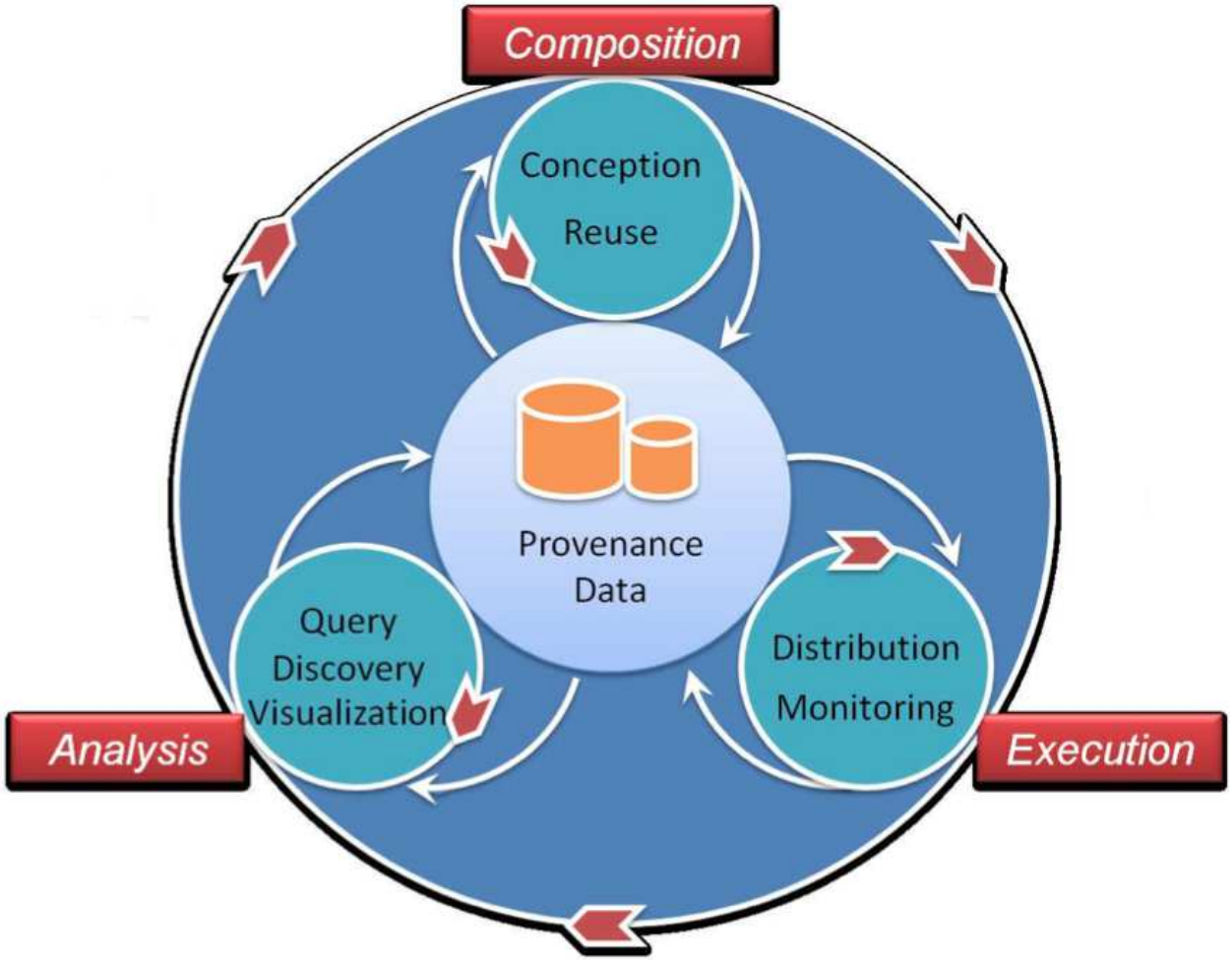
Fonte: Dark Energy Survey (`http://www.darkenergysurvey.org`)

Exemplo: Workflow do DES



K. Kotwani et al. The Dark Energy Survey Data Management System as a Data Intensive Science Gateway. *Proc. of the 8th International Workshop on Middleware for Grids, Clouds and e-Science* (MGC 2010). ACM, 2010.

Ciclo de Vida de Experimentos Científicos



M. Mattoso, C. Werner, G. Travassos, V. Braganholo, E. Ogasawara, D. Oliveira, S. Cruz, W. Martinho, and L. Murta, Towards supporting the life cycle of large scale scientific experiments. *International Journal of Business Process Integration and Management* 5(1):79–92, 2010.

Ciclo de Vida de Experimentos Científicos

- **Composição**. Especificação das atividades e dos dados que serão utilizados. Pode incluir o reuso de especificações de experimentos similares.
- **Execução**. Mapeamento, distribuição e inicialização das atividades do experimento em tarefas concretas executadas em recursos computacionais. Inclui também o monitoramento das mesmas e a movimentação dos dados necessários.
- **Análise**. Avaliação do experimento através dos resultados obtidos, p. ex. dados de saída. Pode se basear também em consultas aos registros de eventos da execução (proveniência).

M. Mattoso et al. Desafios No Apoio à Composição De Experimentos Científicos Em Larga Escala. *Anais do XXXVI Seminário Integrado De Software e Hardware* (SEMISH), *XXIX Congresso Da Sociedade Brasileira De Computacão* (CSBC), p. 307-321, 2009.

Requisitos de Experimentos Científicos Computacionais

- **Abstração**. Deve ser possível ao cientista especificar o experimento em alto nível, sem precisar detalhar aplicações específicas e onde elas serão executadas.
- **Reprodutibilidade**. O experimento pode ser reproduzido por terceiros de forma independente.

  D. Koop et al. A Provenance-Based Infrastructure for Creating Executable Papers. *Proceedings of the ICCS*, 2011.
- **Reutilização**. Deve ser possível reutilizar especificações de experimentos realizados previamente para concepção de novos experimentos.

  Exemplo: MyExperiment (`http://www.myexperiment.org`).
- **Escalabilidade**. O experimento deve ser resiliente ao aumento do número de tarefas e da quantidade de dados.

Experimentos Científicos Computacionais em Larga Escala

- A execução deve ser escalável, ou seja, resiliente ao aumento do número de tarefas e da quantidade de dados.
- Experimentos em larga escala normalmente requerem o uso de computação paralela e distribuída:
  - **Computação paralela**. Utilização de múltiplos processadores de forma concorrente para reduzir o tempo de execução de atividades.

    I. Foster. Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. Addison-Wesley, 1994.
  - **Computação em grade**. Compartilhamento de recursos computacionais heterogêneos e distribuídos.

    I. Foster e C. Kesselman, Eds. The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 2003.
  - **Computacão na nuvem**. Utilização de recursos de terceiros.

    D. Oliveira, F. Baião e M. Mattoso. Towards a Taxonomy for Cloud Computing from an e-Science Perspective. Cloud Computing, pp. 47–62. Springer, 2010.

Escalabilidade de Experimentos Científicos Computacionais

- O tempo total de execução de uma computação paralela e distribuída é dado por:

$$t = t_{\text{processando}} + t_{\text{comunicando}} + t_{\text{ocioso}}$$

- Existem diversas ferramentas para análise de performance de aplicações de forma isolada.
- Um desafio é realizar o mesmo tipo de análise para experimentos científicos computacionais, compostos por muitas aplicações.
- A escalabilidade depende da orquestração da execução de tarefas e gerência de dados de forma eficiente.

Grades Computacionais

- A realização destes experimentos requer o compartilhamento de recursos computacionais distribuídos.
- Computação em grade permite ganhos de escala em termos de processamento.
- Gerência de dados em grades é uma área em desenvolvimento.

Grades Computacionais

- Problemas em gerência de dados distribuída:
  - armazenamento, acesso e replicação dados: SRB, iRODS;
  - gerência de workflows: Kepler, Swift (Globus), Taverna;
  - gerência de metadados: Chimera;
  - processamento de consultas de alto nível: OGSA DAI/DQP.

A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke. The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications*, 23:187-200, 2001.

E. Pacitti, P. Valduriez, M. Mattoso. Grid Data Management: Open Problems and New Issues. *Journal of Grid Computing*, 5(3):273-281, 2007.

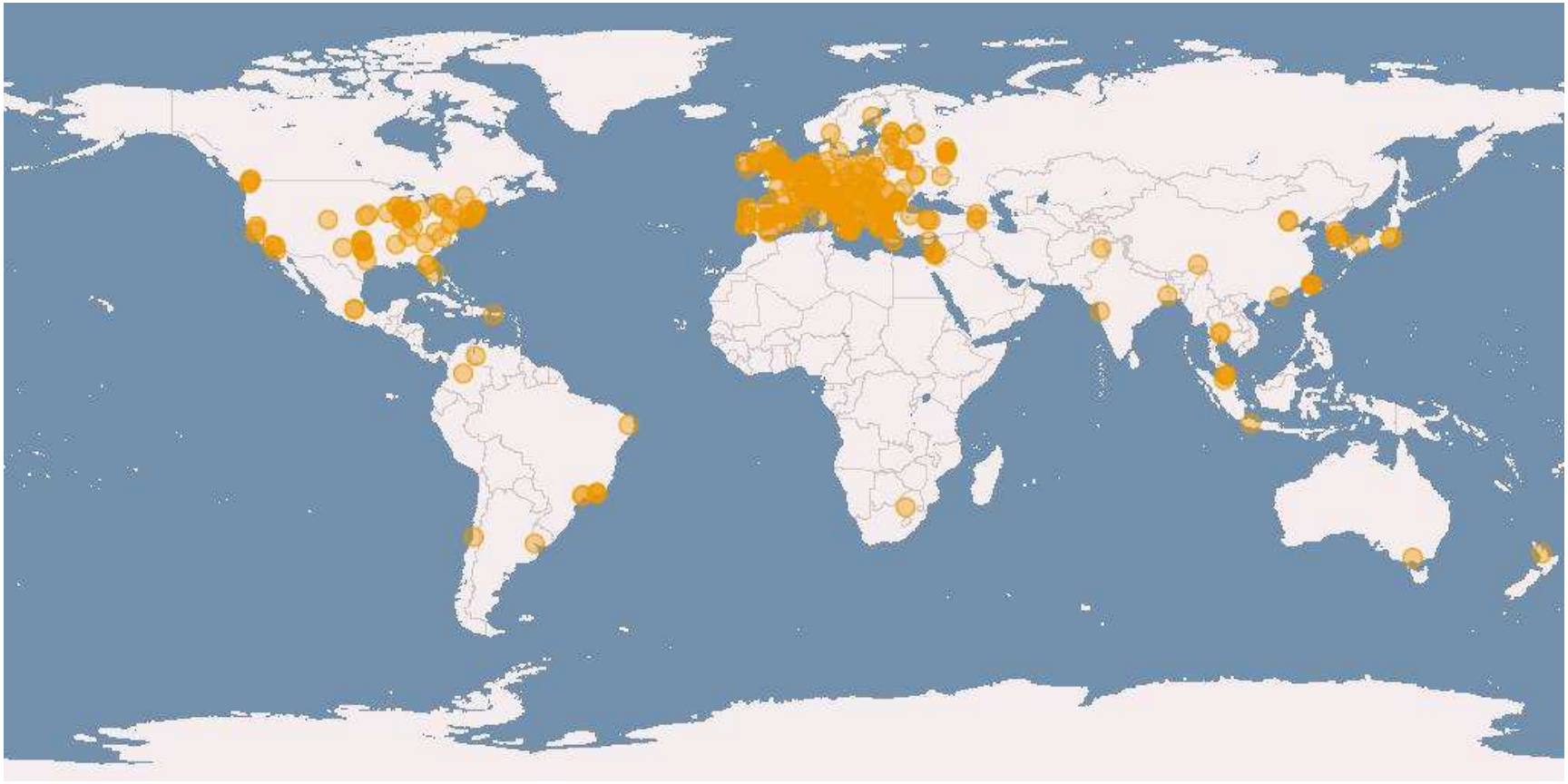
Grades Computacionais

- Uma **grade computacional** é uma infra-estrutura composta de software e hardware que provê recursos computacionais de alto desempenho.
- Tais recursos podem estar distribuídos entre múltiplas instituições sob domínios administrativos distintos.
- O *Globus* é um conjunto de ferramentas para implementação de grades computacionais.

Exemplo: Large Hadron Collider (LHC)

- Colisor de partículas de 27km de circunferência localizado na fronteira da França e Suíça.
- Projeto colaborativo envolvendo cerca de 100 países e 10.000 cientistas.
- Composto por seis detectores que geram cerca de 15PB de dados por ano.
- LHC Computing Grid (LCG):
  - Estrutura de computação distribuída organizada de forma hierárquica, para execução de diferentes partes do workflow do experimento:
    - Tier 0: processamento de dados brutos dos detectores,
    - Tier 1: reconstrução e pré-processamento,
    - Tier 2: simulação e análise.

Exemplo: Mapa do LCG

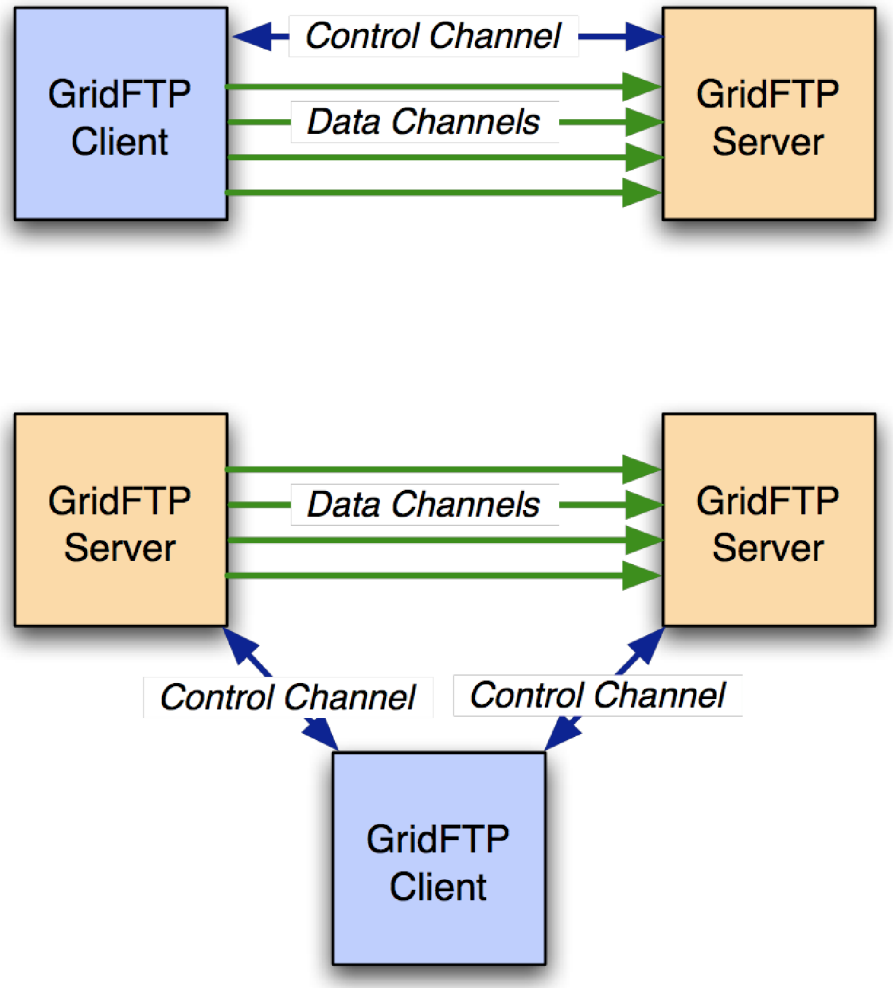


Fonte: CERN (`http://gstat-prod.cern.ch`)

Grades Computacionais

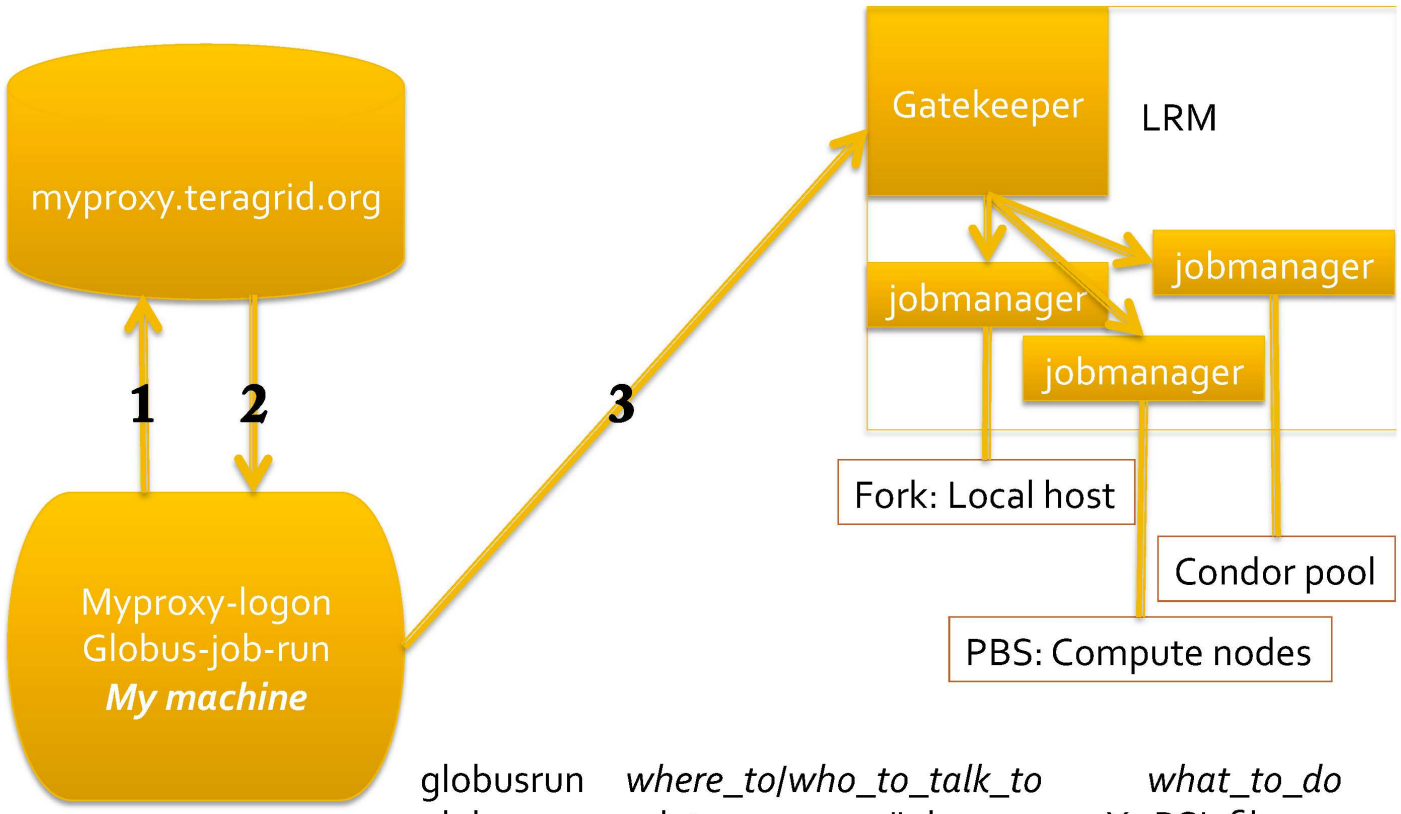
- Principais serviços:
  - Globus Resource Allocation Manager (GRAM): gerenciador de recursos computacionais, recebe submissão de tarefas e as repassa para o gerenciador de execução local (*fork*, fila de execução).
  - GrifFTP: transferência de arquivos entre domínios administrativos.

GridFTP



Fonte: XSEDE (`http://www.xsede.org`)

GRAM



globusrun *where_to/who_to_talk_to* *what_to_do*
globusrun url_to_resource/jobmanager-X RSL file

Fonte: XSEDE (`http://www.xsede.org`)

- Questões:
  - Como gerenciar identidades através de domínios administrativos?
  - A *Grid Security Infrastructure* (GSI) utiliza técnicas de criptografia e certificação digital para gerenciar identidades através de domínios administrativos com segurança.

Workflows Científicos

- Um **workflow científico** consiste da especificação de um conjunto de aplicações científicas a serem executadas e suas dependências mútuas.
- Segue um ciclo de vida análogo ao dos experimentos científicos computacionais:
  - Composição, representação e modelagem de dados.
  - Mapeamento e execução.
  - Coleta de metadados e proveniência.
- Um **sistema de gerência de workflows científicos** (SGWC) permite gerenciar o ciclo de vida de workflows científicos.

E. Deelman et al. Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems* 25(1):528-540, 2009.

Workflows Científicos

- **Sistemas de gerência de workflows científicos** (SGWC) visam a automação de experimentos científicos computacionais:
  - escalonamento de tarefas baseado em dependências de dados;
  - fluxo de dados entre tarefas;
  - execução paralela de tarefas independentes;
  - escalonamento de tarefas em ambientes de computação de alto desempenho;
  - gerência e consulta de dados de proveniência.

Exemplo: Laboratório de Bioinformática, LNCC

Sequenciador Roche 454 FLX gera 12GB a 15GB por rodada de sequenciamento.
Dados processados por várias aplicações:

- filtragem por qualidade de dados gerados pelo sequenciador.
- formatação de dados.
- comparação das sequências com bases de dados conhecidas.

Exemplo: Laboratório de Bioinformática, LNCC

Laboratório Nacional de Computação Científica

Composição de Workflows Científicos

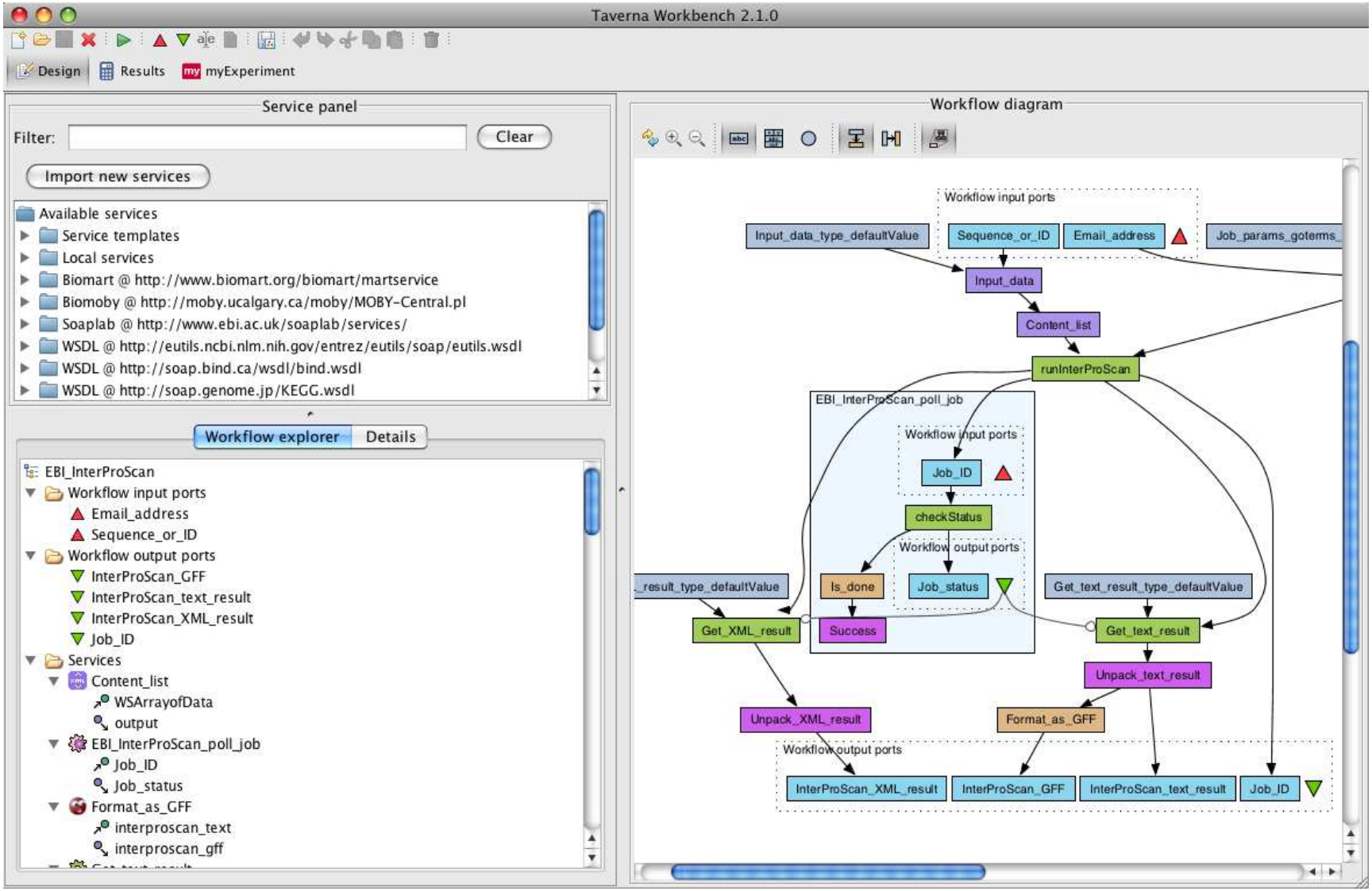
- Etapa em que são definidas as aplicações componentes e as dependências de dados.
- Níveis de especificação:
  - Abstrato: tarefas abstratas, descritas por funcionalidade geral (p. ex., comparação de seqüências).
  - Concreto: aplicações científicas e conjuntos de dados específicos.
- Representação:
  - Textual: linguagem de programação.
  - Gráfica: interface gráfica onde nós são tarefas e arestas são dependências.

Exemplo: Composição Textual

```
type messagefile;
type countfile;

app (countfile t) countwords (messagefile f) {
     wc "-w" @filename(f) stdout=@filename(t);
}

string inputNames = "foreach.1.txt foreach.2.txt foreach.3.txt";

messagefile inputfiles[] <fixed_array_mapper;files=inputNames>;

foreach f in inputfiles {
  countfile c <regexp_mapper; source=@f, match="(.*)txt",
               transform="\\1count">;
  c = countwords(f);
}
```

Workflow especificado em Swift (`http://www.ci.uchicago.edu/swift`)

Exemplo: Composição Gráfica

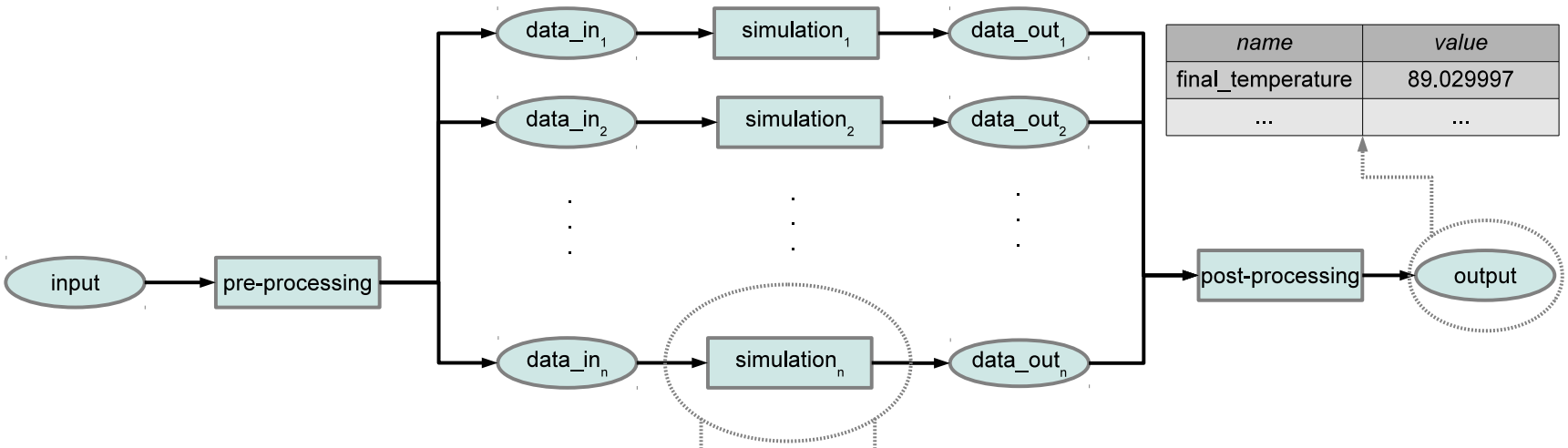


Fonte: Taverna (`http://www.taverna.org.uk`)

Coleta de metadados e proveniência

- Coleta de eventos do ciclo de vida do workflow:
  - Mudanças e evolução da especificação.
  - Consumo e produção de dados por aplicações componentes executadas.
- Gerência de metadados relacionados ao domínio científico (semântica do experimento).
- Serviço de consulta a metadados e proveniência.
- Aplicações: reprodutibilidade, verificação, análise.

Coleta de metadados e proveniência



name	*value*
final_temperature	89.029997
...	...

name	*value*
execution_site	pads.ci.uchicago.edu
version	1.15
...	...

timestamp	*mem_usage*	*cpu_load*	*io_read*	*io_write*
1314104905	126342	99	230	0
1314104906	155739	100	123	1044
...	...	...	...	...
1314112105	394523	99	0	129

Exemplos de SGWCs: Swift

- Permite gerenciar workflows científicos em ambientes paralelos e distribuídos.
- É composto por:
  - Uma linguagem de alto nível para especificação de workflows científicos como scripts.
  - Ambiente de execução com suporte nativo a:
    - execução local,
    - execução paralela (p.ex. PBS, SGE),
    - execução distribuída (p.ex. Condor, Globus).
  - Sistema de gerência de proveniência.

Swift (`http://www.ci.uchicago.edu/swift`)

Exemplos de SGWCs: SciCumulus

- Permite gerenciar workflows científicos em ambientes de nuvem.
- É composto por:
  - Especificação de workflows em XML.
  - Motor de execução baseado em álgebra relacional.
  - Sistema de gerência de proveniência.

SciCumulus (`http://sourceforge.net/projects/scicumulus/`)

Exemplos de SGWCs: Taverna

- Aplicações componentes podem ser serviços web.
- Integração com o portal de reutilização de workflows MyExperiment.
- Especificação de workflows através de interface gráfica.
- Suporte a coleta e consultas de proveniência.
- Popular na área de bioinformática.

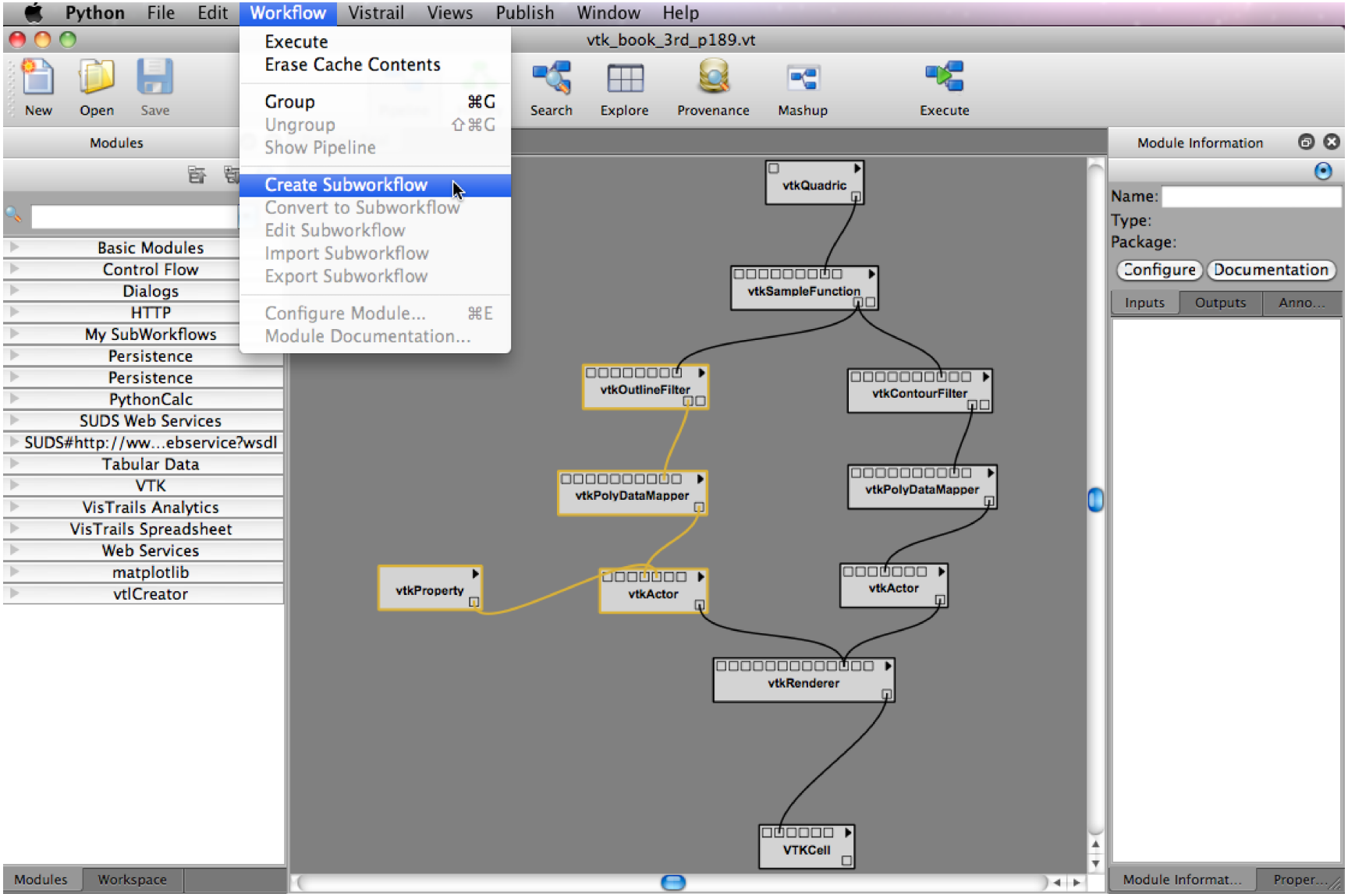
Taverna (`http://www.taverna.org.uk`)

Exemplos de SGWCs: Vistrails

- Suporte a versionamento e gerência da evolução de workflows.
- Especificação de workflows através de interface gráfica.
- Suporte a coleta e consultas de proveniência.
- Popular na área de visualização científica.

Vistrails (`http://www.vistrails.org`)

Exemplos de SGWCs: Vistrails



Fonte: Vistrails (`http://www.vistrails.org`)

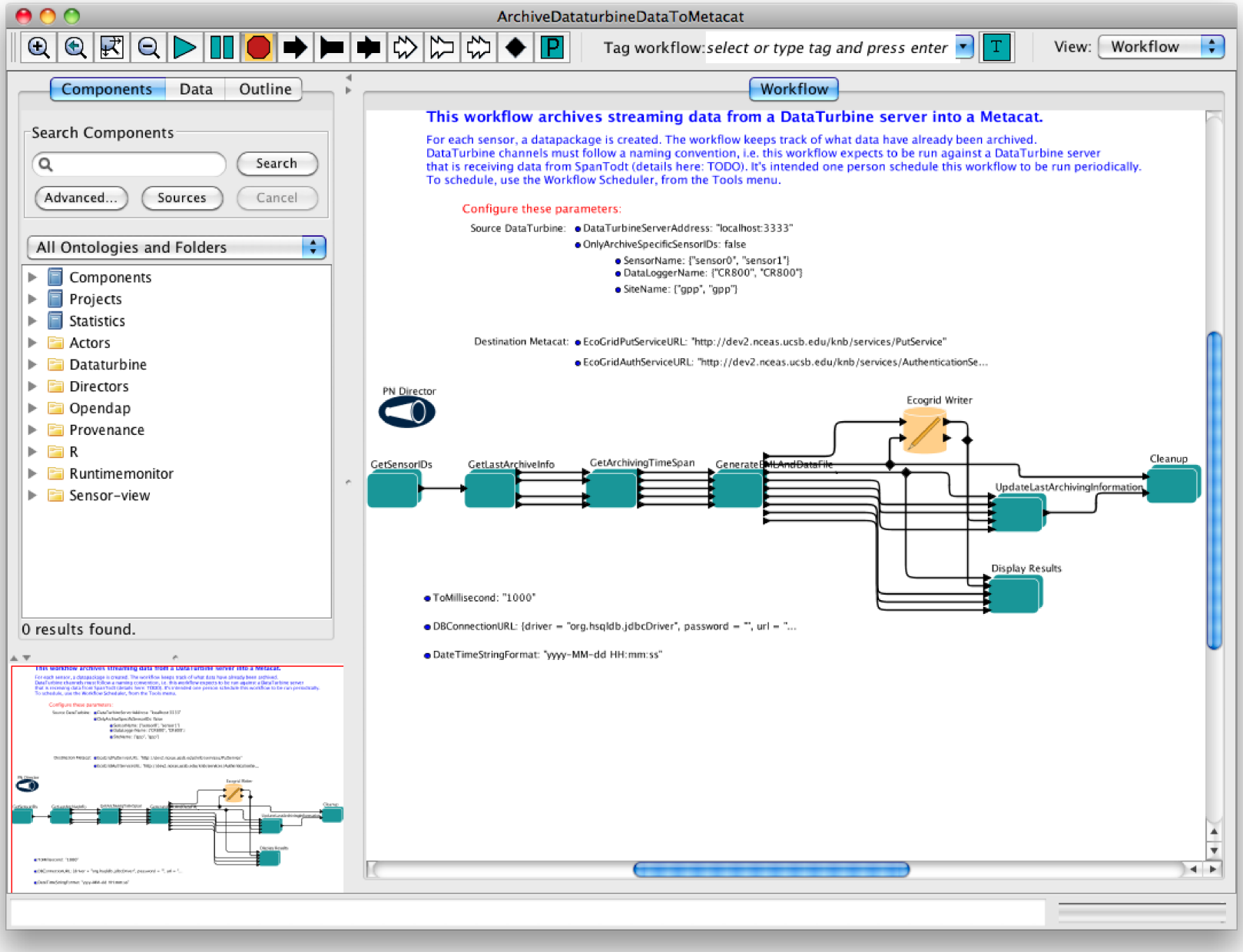
Laboratório Nacional de Computação Científica

Exemplos de SGWCs: Kepler

- Suporte a aplicações disponibilizadas através de serviços web.
- Especificação de workflows através de interface gráfica.
- Suporte a coleta e consultas de proveniência.
- Integração com ferramentas populares, como o R.

Kepler (`https://kepler-project.org`)

Exemplos de SGWCs: Kepler



Ambiente de recomendado para prática

- Laboratório 5
- Recursos do CENAPAD-RJ:
  - Cluster Sun Blade X6250:
    - 78 nós com 2 processadores quad-core Intel Xeon E5440 e 16GB de memória.
    - Total de 624 núcleos de processamento e 1.24TB de memória.
    - URL: `http://www.lncc.br/sunhpc`.
  - Cluster SGI Altix XE:
    - 30 nós com 2 processadores quad-core Intel Xeon E5520 e 24GB de memória.
    - Total de 240 núcleos de processamento e 720GB de memória.
    - URL: `http://www.lncc.br/altix-xe`.
- URL: `http://www.cenapad-rj.lncc.br`.