A Framework for Distributed Data-Parallel Execution in the Kepler Scientific Workflow System

Jianwu Wang, Daniel Crawl, Ilkay Altintas
San Diego Supercomputer Center
University of California, San Diego

http://biokepler.org/
Background

• Scientific data
  – Enormous growth in the amount of scientific data
  – Applications need to process large-scale data sets

• Data-intensive computing
  – Distributed data-parallel (DDP) patterns, e.g., PACT and MapReduce, facilitate data-intensive applications
  – Increasing number of execution engines available for these patterns, such as Hadoop and Stratosphere
Challenges

- Applications or workflows built using these DDP patterns are usually **tightly-coupled** with the underlying DDP execution engine.
- None of existing applications/systems support workflow execution on **more than one** DDP execution engine.
The bioKepler Approach

• Use Distributed Data-Parallel (DDP) frameworks, e.g., MapReduce, to execute bioinformatics tools

• Create configurable and reusable DDP components in Scientific Workflow System

• Support different execution engines and computational environments

http://biokepler.org/
Conceptual Framework

Bioinformatics Tools
- Mapping
- Clustering
- Assembly

Bioinformatician
- Customize & Integrate

Data
- Ensembl
- CAMERA
- Genbank

Transfer

Compute
- Amazon EC2
- Triton Resource
- FutureGrid
- Sun Grid Engine

Adhoc Network

Kepler
- Data-Parallel Execution Patterns
  - Map-Reduce
  - Master-Slave
  - All-Pairs

bioKepler
- Director
  - Scheduler
  - Executable Workflow Plan
  - Execution Engine

Deploy & Execute

Provenance
- Data Lineage
- Execution History

Reporting
- Report Designer
- PDF Generation

Run Manager
- Search
- Tag

Fault-Tolerance
- Alternatives
- Error Handling

06/02/12
http://biokepler.org/
bioKepler Architecture

Kepler Scientific Workflow System
- Provenance
- Fault-Tolerance
- Reporting
- Distributed Data-Parallel
  - Directors
    - Stratosphere
    - Hadoop
  - Actors
    - Patterns: Map, Reduce, ...
    - Data I/O: HDFS, local, ...
- Actors
  - SQL, REST
  - R, Matlab
  - ...
  - Directors
    - SDF, PN, ...

DDP Execution Engines
- Hadoop
- Stratosphere
- ...

Computational Environments
- Cloud
- Grid
- Cluster
- ...

06/02/12
http://biokepler.org/
Distributed Data-Parallel bioActors

• **Set of steps to execute a bioinformatics tool in DDP environment**
  
• **Can either be:**
  – as sub-workflows (composite)
  – in Java code (atomic)

• **Includes:**
  – Data-parallel patterns, e.g., Map, Reduce, All-Pairs, etc. to specify data grouping
  – I/O to interface with storage
  – Data format specifying how to split and join
Distributed Data-Parallel Directors

- Directors implement a *Model of Computation*
  - Specify when actors execute
  - How data transferred between actors
- DDP Directors run bioActors on DDP execution engines
  - Hadoop director converts workflow into MapReduce, runs on Hadoop
  - Stratosphere director converts workflow into PACT program, executes on Nephele
  - Generic DDP director automatically detect available DDP engines and select the best
data partition for each execution

DDP BLAST Workflow

Data partition for each execution...
DDP bioActor Usage Model
DDP BLAST Workflow Experiments
Summary

• **The bioKepler approach**
  – Facilitates using data-parallel patterns for distributed execution of bioinformatics tools
  – Interfaces with different execution engines to use various computational resources

• **Future Work**
  – Which patterns for which tools?
  – New patterns needed?
Questions?

• More Information

{jianwu, crawl, altintas}@sdsc.edu
http://www.kepler-project.org
http://www.bioKepler.org

• Acknowledgements
  – NSF OCI-0722079 for Kepler/CORE, DBI-1062565 for bioKepler
  – Gordon and Betty Moore Foundation for CAMERA
  – UCSD Triton Research Opportunities Grant