Exploring Architecture Options for a Federated, Cloud-based Systems Biology Knowledgebase

Ian Gorton, Jenny Liu, Jian Yin
Systems Biology

- Integrated study of organisms as a whole
- Obtain, integrate, and analyze complex data from multiple experimental sources using interdisciplinary tools

Requirements

- Large amount of data
- Different types of tools
- Large amount of computation resources
Systems Biology Knowledgebase

- **Drawbacks of the current approach**
  - Threshold of entrance can be high
  - Little reusing and sharing of the data and tools, wasteful repetitive effort to develop similar software tools
  - Results are hard to replicated

- **Seamlessly sharing and integration of data and software tools between multiple institution are attractive**

- **The goal of system biology knowledgebase is to exploit cloud computing technologies to enable sharing of data and software tools**
Why Cloud Computing

- Enable sharing of data and software tools
- Dynamic allocation of computing resources
- Many software tools can be converted to run on top of cloud computing services such as Hadoop
Outline

- Introduction
- System Architecture
- Prototype of selected components
  - Case study
  - Hadoop based systems biology tools
- Conclusion
Centralize verse Federated

- Advantages of centralized approach
  - Ease of integration
  - More efficient computing resource allocations

- However, many institute may want to retain controls of their data and tools

- Federated approach
  - Leverage specialized computing resources across organizations
Architecture Overview

Workflow Tools, Web Portals, Desktop Apps

- cURL
- php
- java
- python
- scripts

Middleware and Workflow Utilities

RESTful API Layer

- Database Adaptors
- Data and Resource Directories

Kbase Interface Layer

(for flexible federation of Kbase Data and Compute resources)

Semantic Access Interface Layer

Cloud-based data APIs

Cloud storage
  - e.g. S3

Cloud computations
  - e.g. EC2

HPC-based computations
  - e.g. Clusters

Kbase core

User Access Layer

Infrastructure Layer

Federation Layer

Example Federated Resources

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965
Components

- Location independent components
- Uniformed interfaces
- Easy composition
- Execution can be monitored with JBPM
Secure Communication

- Security must be ensured for communication across institutions
  - Only SSL traffic are allowed through firewall
- Requiring all the components to use SSL could be difficult
- Use SOCKS to minimize code changes of components
Example

► Original code

```java
URL url = new URL(urlname);
```

► Modified code

```java
SocketAddress addr = new InetSocketAddress("localhost", 8182);
Proxy proxy = new Proxy(Proxy.Type.SOCKS, addr);
URL url = new URL(urlname);   // Create the URL
URLConnection uc = url.openConnection(proxy);
```
Advanced Visualizations
VESPA

Prototype

- **Polygraph**
  - Protein fasta file (.faa file)
  - Script: translate DNA (.fna file) in six frames
  - Query & copy the .fna file
  - Query & copy the .faa file
  - post-process script

- **Polygraph**
  - parameter file
  - Query & copy the .dta files
  - peptide file

- **GenBank**
  - Query & copy the .fna file and the .gbk file

- **Visualization tool**
  - Visualization at a user’s local workstation

- **Proteomics data (dta files)**
  - Proteins fasta file (.faa file)

Proudly Operated by Battelle Since 1965
Polygraph is a proteomics application to identify peptides from MS data
Initially implemented with MPI
Loosely coupled and suitable for Hadoop
Small amount of effort to adapt it to run on top of Hadoop
### Experimental Results

<table>
<thead>
<tr>
<th># of Mappers</th>
<th>PolyGraph Runtime (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2637</td>
</tr>
<tr>
<td>20</td>
<td>1322</td>
</tr>
<tr>
<td>30</td>
<td>897</td>
</tr>
<tr>
<td>50</td>
<td>565</td>
</tr>
<tr>
<td>100</td>
<td>291</td>
</tr>
<tr>
<td>200</td>
<td>158</td>
</tr>
</tbody>
</table>
MPI-base implementation is highly tuned and thus more efficient

Hadoop based approach is more flexible
- Most cloud computing providers provide Hadoop service
- Flexibility for leveraging various amounts of computing resource without changing code
  - Can produce results even with one machine
  - More machines can speed up the computation

Many system biology applications can be adapted to Map Reduce paradigm
Conclusion

- Sharing data, software tools, and computing resources is essential for systems biology
- Cloud computing can provide the ideal platforms
  - Many applications are loosely coupled and can be adapted to run in cloud computing environments
- Federated approach provides more flexibility
- Uniformed interfaces enable easy integration