An Interactive Workflow Generator to Support Bioinformatics Analysis through GPU Acceleration

Anuradha Welivita, Indika Perera, Dulani Meedeniya
Department of Computer Science and Engineering
University of Moratuwa, Sri Lanka
Outline

➔ Introduction
➔ Literature Review
➔ Methodology
➔ Implementation
➔ Evaluation and Results
➔ Conclusions
Introduction
Bioinformatics analyses play a significant role in Bioinformatics research. Carried out by constructing pipelines that executes multiple software tools in a sequential fashion. Workflow systems generated to simplify the construction of pipelines and automate analyses.
Biological data is ever increasing
  - Hence, difficult to get results within reasonable period of time

GPU accelerated computing has now become the mainstream for HPC applications

But currently available solutions only provide distributed system support for parallelized computations
  - E.g. Galaxy, Taverna
Project Objectives

- An interactive workflow generation system
- Analyses through cloud based GPU computing resources
- Supporting specific requirements of bioinformatics software
Existing Techniques

➡️ Scripting
  ○ A low level, a less abstract method of using basic scripting languages to generate workflows.

➡️ Makefiles
  ○ A script having a set of rules defining a dependency tree declaratively.

➡️ Scientific workflow management systems
  ○ Software that provides an infrastructure to set up, execute, and monitor workflows.
## Evaluation of Existing Techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Advantages</th>
<th>Limitations</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scripting</td>
<td>➔ Simple to construct</td>
<td>➔ Not support for shared file systems Bash, Perl, Python</td>
<td>➔ Bash, Perl, Python</td>
</tr>
<tr>
<td></td>
<td>➔ Openness</td>
<td>➔ Development overhead</td>
<td></td>
</tr>
<tr>
<td></td>
<td>➔ Ability to execute from command line</td>
<td>➔ Hard to determine the exact point of failure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>➔ Extreme flexibility to manipulate pipelines</td>
<td>➔ Difficult to reproduce analyses</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>➔ Difficult to integrate new tools and databases</td>
<td></td>
</tr>
</tbody>
</table>
# Evaluation of Existing Techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Advantages</th>
<th>Limitations</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Makefiles</td>
<td>➔ Simple to construct</td>
<td>➔ Not flexible compared to scripting Make, CMake</td>
<td>➔ Make</td>
</tr>
<tr>
<td></td>
<td>➔ Describe the data flow</td>
<td>➔ Single wild-card per rule restriction SCanS</td>
<td>➔ CMake</td>
</tr>
<tr>
<td></td>
<td>➔ Take care of dependency resolution</td>
<td>➔ Cannot describe a recursive flow Makeflow</td>
<td>➔ SCanS</td>
</tr>
<tr>
<td></td>
<td>➔ Commands can be executed in parallel</td>
<td>➔ Require programming or shell experience Snakemake</td>
<td>➔ Makeflow</td>
</tr>
<tr>
<td></td>
<td>➔ Cache results from previous runs</td>
<td>➔ Deceptive error messages</td>
<td>➔ Snakemake</td>
</tr>
<tr>
<td></td>
<td>➔ State dependencies among files &amp; commands</td>
<td>➔ No support for multi-threaded/multi-process jobs</td>
<td></td>
</tr>
</tbody>
</table>
### Evaluation of Existing Techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Advantages</th>
<th>Limitations</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scientific Workflow Management Systems</td>
<td>➔ Interconnects components</td>
<td>➔ Require more effort</td>
<td>➔ Galaxy</td>
</tr>
<tr>
<td></td>
<td>➔ Do not require programming experience</td>
<td>➔ No authority to standardize for interoperability</td>
<td>➔ Taverna</td>
</tr>
<tr>
<td></td>
<td>➔ Enable reproducible data analysis</td>
<td></td>
<td>➔ Bioconductor</td>
</tr>
<tr>
<td></td>
<td>➔ Can simply integrate with HPC systems</td>
<td></td>
<td>➔ BioPython</td>
</tr>
<tr>
<td></td>
<td>➔ Allow execution on distributed resources</td>
<td></td>
<td>➔ Nextflow Workbench</td>
</tr>
</tbody>
</table>
Scientific Workflow Management Systems

Galaxy
# Galaxy

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>➔ Support reproducibility of results and transparency of workflow execution</td>
<td>➔ Does not enable GPU based computations</td>
</tr>
<tr>
<td>➔ Available to users via a simple web interface</td>
<td></td>
</tr>
<tr>
<td>➔ Provides distributed computing support for calculations</td>
<td></td>
</tr>
</tbody>
</table>
Scientific Workflow Management Systems

→ Taverna
## Taverna Workbench

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>➔ Enables integration of tools distributed across the internet</td>
<td>➔ Being available only as a stand-alone application makes it less accessible by the community</td>
</tr>
<tr>
<td>➔ Provides a web based platform for sharing workflows</td>
<td>➔ Limited by the platform it runs on</td>
</tr>
<tr>
<td>➔ Provides distributed computing support for calculations</td>
<td>➔ Does not enable GPU based computations</td>
</tr>
</tbody>
</table>
Challenges and Limitations

- Large-scale data-intensive bioinformatics analyses pose significant challenges on performance and scalability.

- Currently available solutions only provide distributed system support for parallelized computations.
  - E.g. Galaxy, Taverna

- But use of GPUs in the cloud can harness the power of GPU computation from the cloud itself and on demand.
Process

Review

Develop

Evaluate
Web App Development

➡ SPA developed using JavaScript and NodeJS
➡ Front end development using AngularJS
➡ Hosted in an Amazon EC2 P2 instance
  ○ A GPU accelerated cloud platform
  ○ With up to 16 NVIDIA Tesla K80 GPUs
  ○ Scalable and provides parallel computing capabilities
High Level Architecture

Web application frontend

Views

Controllers

D3.js

jsPlumb

Frontend Services

Node Server

Backend Database

External tools and Datasets

Tools (Binaries) / Web Services

Datasets
Implementation Details
Enhancing Performance

→ How GPU acceleration works in a simple workflow
Amazon EC2 P2 virtual machine instances

- Amazon EC2 - a cloud based instance that enables hosting of HPC applications
- Amazon EC2 P2 - a type of Amazon EC2 cloud instance that supports computations on NVIDIA k80 GPUs
Enhancing Performance

- On-demand scaling across a cluster of nodes
  - Achieved through Amazon Elastic Load Balancing (ELB)
Implementation of Specific Requirements

- Interactive and Graphical Workflow Creation
- Module Extensibility
- Reporting
- Reproducibility
- User Management
1) Interactive & Graphical Workflow Creation
2) Module Extensibility

- Capability to add/remove data processing or analysing components
- A plugin architecture for service addition
- Add/remove tools & services via updating a JSON configuration file
2) Module Extensibility

```json
{
    "WebServicesList": [
        {
            "Name": "ncbi",
            "List": [
                {
                    "Id": "S007",
                    "Name": "ElastN",
                    "InputParams": [
                        { "name": "db", "type": "db", "value": "" },
                        { "name": "query", "type": "seq", "value": "" }
                    ],
                    "OutputParam": { "output": "" },
                    "Description": "BlastN compares nucleotide sequences by local alignment"
                }
            ]
        }
    ],
    "Desc": "List of web services offered by ncbi"
}
```
3) Reporting

→ Helps maintaining details of the executed pipelines and summaries of analysis
→ Automatic HTML report generation using PhantomJS
→ D3.js to generate analysis specific visualizations
3) Reporting

→ Analysis specific results visualization
4) Reproducibility

→ Challenges

○ Original data may be modified or deleted by the researchers.
○ Data may get corrupted by transfer processes.
○ Versions of software tools change, services become unavailable or software used may become proprietary.
4) Reproducibility

Solution

- Maintain an audit trail with technical metadata.
- A separate thread to record details each time the workflow is updated.
- Each step recorded in a backend relational database and accessed whenever the workflow needs to be reproduced.
5) User management

- Importance of having a proper user management and authentication system,
  - To track and share individual analyses
  - To keep track of user data
  - To process quotas
5) User management

- Amazon Cognito
  - User registration & authentication
  - Data synchronization
### Summary of comparison of features with existing systems

<table>
<thead>
<tr>
<th>Feature</th>
<th>Taverna</th>
<th>Galaxy</th>
<th>BioFlow</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Performance</strong></td>
<td>Computation on distributed computing environments</td>
<td>Use of Amazon cloud and local grid support to distribute workload</td>
<td>Enhanced performance using GPU accelerated Amazon cloud services</td>
</tr>
<tr>
<td><strong>2. Interactive graphical workflow creation</strong></td>
<td>GUI based workbench</td>
<td>A web based graphical workflow editor</td>
<td>A web based GUI for workflow generation on HTML canvas.</td>
</tr>
</tbody>
</table>
## Summary of comparison of features with existing systems

<table>
<thead>
<tr>
<th>Feature</th>
<th>Taverna</th>
<th>Galaxy</th>
<th>BioFlow</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Module Extensibility</td>
<td>➔ Computation on distributed computing environments</td>
<td>➔ Use of Amazon cloud and local grid support to distribute workload</td>
<td>➔ Enhanced performance using GPU accelerated Amazon cloud services</td>
</tr>
<tr>
<td>4. Reporting</td>
<td>➔ GUI based workbench</td>
<td>➔ A web based graphical workflow editor</td>
<td>➔ A web based GUI for workflow generation on HTML canvas.</td>
</tr>
<tr>
<td></td>
<td>➔ Poor drag &amp; drop of workflow items</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Summary of comparison of features with existing systems

<table>
<thead>
<tr>
<th>Feature</th>
<th>Taverna</th>
<th>Galaxy</th>
<th>BioFlow</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5. Reproducibility</strong></td>
<td>➔ Computation on distributed computing environments</td>
<td>➔ Use of Amazon cloud and local grid support to distribute workload</td>
<td>➔ Enhanced performance using GPU accelerated Amazon cloud services</td>
</tr>
<tr>
<td><strong>6. User management</strong></td>
<td>➔ GUI based workbench</td>
<td>➔ A web based graphical workflow editor</td>
<td>➔ A web based GUI for workflow generation on HTML canvas</td>
</tr>
<tr>
<td></td>
<td>➔ Poor drag &amp; drop of workflow items</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluation and Results
Performance Evaluation

Workflow run on top of a GPU enabled Amazon EC2 Linux instance, having the following CPU and GPU specifications.

<table>
<thead>
<tr>
<th>CPU</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel(R) Core(TM) i5-2450M CPU</td>
<td>Nvidia GeForce GT 525M</td>
</tr>
<tr>
<td>2 cores, @ 2.50 GHz</td>
<td>96 CUDA Cores</td>
</tr>
<tr>
<td>4 GB RAM</td>
<td>1 GB RAM</td>
</tr>
</tbody>
</table>
Performance Evaluation

- Workflow executed by inputting different lengths of query sequences
- Used both remotely installed ncbi-blast and GPU-Blast
- GPU-Blast
  - Accelerate gapped & ungapped protein sequence alignments
Performance Evaluation

Evaluation results

<table>
<thead>
<tr>
<th>Length of input seq.</th>
<th>Time on CPU (sec.)</th>
<th>Time on GPU (sec.)</th>
<th>Speedup ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.019</td>
<td>0.017</td>
<td>1.12</td>
</tr>
<tr>
<td>300</td>
<td>0.023</td>
<td>0.018</td>
<td>1.28</td>
</tr>
<tr>
<td>600</td>
<td>0.050</td>
<td>0.020</td>
<td>2.50</td>
</tr>
<tr>
<td>900</td>
<td>0.060</td>
<td>0.021</td>
<td>2.86</td>
</tr>
<tr>
<td>1200</td>
<td>0.073</td>
<td>0.024</td>
<td>3.04</td>
</tr>
<tr>
<td>1500</td>
<td>0.081</td>
<td>0.026</td>
<td>3.11</td>
</tr>
</tbody>
</table>
Performance Evaluation

→ Comparison between execution times on CPU vs GPU
Performance Evaluation

→ Avg. GPU speedup for different lengths of input query seq.
Performance Evaluation

Observations

- When input length increases, speedup ratio also increases.
- Significant increase in performance cannot be observed when the query length is small.
- In long input queries, about 3 fold increase in performance can be obtained through GPU acceleration.
Usability Evaluation

- System Usability Scale (SUS)
  - Subjects: 10 subjects aged 20-30 having basic knowledge in computing and bioinformatics
  - Compared against Taverna Workbench
- Open-ended interview
SUS Scores

72.5%
Taverna Workbench

77.5%
BioFlow

Diagram featured by http://slidemodel.com
Inability to drag individual components in Taverna system makes it inflexible to visualize the workflow the way we want.

As the Taverna system is desktop based, it has certain dependencies to be pre-installed in the user’s local machine, which makes it cumbersome to use.
Conclusions
Research Outcomes

- An interactive workflow generation software
- Ability to process massive datasets in parallel
- Significant increase in speed of execution
- More desirable features of usability
- Attracts users with minimal programming experience
- Significant increase in speed of execution
Project Demo
Future Extensions

→ Exploring applicability of Amazon EC2 FPGA based computing instances to create custom hardware accelerations for the application

→ Inclusion of more features,
  ○ Sharing of workflows
  ○ Pipeline comparison
  ○ Citation support

→ Development of comprehensive user support and interface enhancements
The Team

Anuradha Welivita
Lecturer (Contract)
Dept. of Comp. Sci. & Eng.
University of Moratuwa
Sri Lanka

Dr. Indika Perera
Senior Lecturer
Dept. of Comp. Sci. & Eng.
University of Moratuwa
Sri Lanka

Dr. Dulani Meedeniya
Senior Lecturer
Dept. of Comp. Sci. & Eng.
University of Moratuwa
Sri Lanka

Anuradha Wickramarachchi
Undergraduate
Dept. of Comp. Sci. & Eng.
University of Moratuwa
Sri Lanka

Vijini Mallawarachchi
Undergraduate
Dept. of Comp. Sci. & Eng.
University of Moratuwa
Sri Lanka
Thanks!

Any questions?

You can find me at:
@AnuradhaKW
anuradha@cse.mrt.ac.lk