Makeflow for Bioinformatics

Andrew Thrasher, Irena Lanc, Douglas Thain, Scott Emrich
Department of Computer Science and Engineering
University of Notre Dame

Abstract

Summary: The development of high-throughput sequencing platformssuch as ABI SOLID, Illumina and 454 Life Sciences has made it trivial to acquire billions of base pairs in a matter of days. For alignment tools to keep pace with sequencing technology, they must utilize the computing resources of multiple machines.We apply Makeflow to the SHRiMP (Rumble et al., 2009) and SSAHA (Ning et al., 2001) alignment packages to simplify parallelization and reduce computation time.When combined with batch systems, quick and scalable solutions can be implemented without learning complex programming languages or details of distributed systems.

Availability: Makeflow and example Perl scripts are available at

http://www.cse.nd.edu/ ccl/software/makeflow/.

What is Makeflow?

Makeflow is a tool designed by the Cooperative Computing Lab at Notre Dame.

It has a syntax similar to the Unix tool *Make.* It utilized the Directed Acyclic Graph (DAG) abstraction to express a workflow that can then be run on a variety of computational resources, including Condor, SGE, Workqueue or Unix.

Example:

```
part1 part2 part3: input.data split.py
	./split.py input.data

out1: part1 mysim.exe
	./mysim.exe part1 >out1

out2: part2 mysim.exe
	./mysim.exe part2 >out2

out3: part3 mysim.exe
	./mysim.exe part3 >out3

result: out1 out2 out3 join.py
	./join.py out1 out2 out3 > result
```

Where is Makeflow used?

We have utilized Makeflow as the underlying engine for a Bioinformatics web portal at Notre Dame known as Biocompute. Biocompute currently offers the BLAST, SSAHA and SHRiMP alignment tools to researchers. These tools utilize our 1000 node campus Condor grid to run the workloads in parallel.

Also many members of the Notre Dame Bioinformatics Lab utilize Makeflow in their research.

How is Makeflow used?

We have found it easier to write a Perl script to generate Makeflows. Therefore we primarily utilize Perl scripts to generate a Makeflow for each new batch of computation.

We have utilized Makeflow to take advantage of the convenient parallelism available in many bioinformatics workloads, primarily many sequence alignment. Additionally we have used the workflow nature of Makeflow to automate complex analysis pipelines.

Why is Makeflow used?

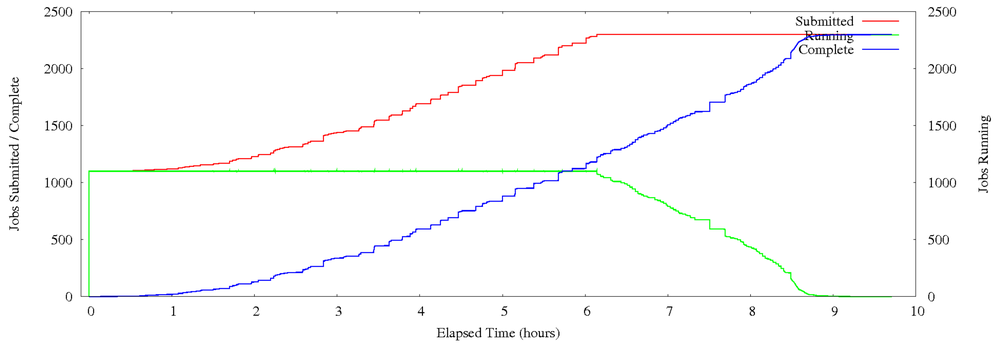
Makeflow abstracts the parallel computing difficulties found in many alternative tools. It simply requires a set of rules stating targets, dependencies and executables. Makeflow creates a DAG and determines which components of the workload are able to be run in parallel and also takes responsibility for transferring the appropriate files.

Makeflow provides an easy to use alternative tool for accomplishing bioinformatics tasks in a distributed computing environment. It is flexible because it only requires each step to be a Unix command line executable.

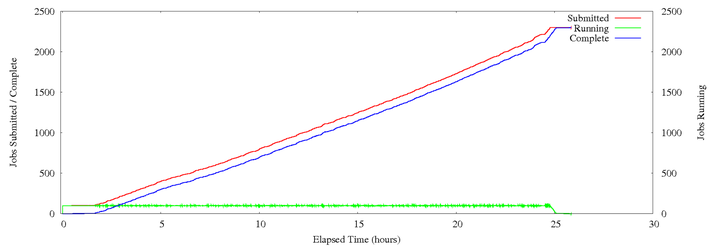
Results

Run times for each workload using 100 processors

Workload	Run time	Total CPU time	Speedup
A. gambiae M form	2 hours	7 days	80x
Oryza rufipogon	3 hours	11 days	86x
S. bicolor	17 hours	67 days	95x

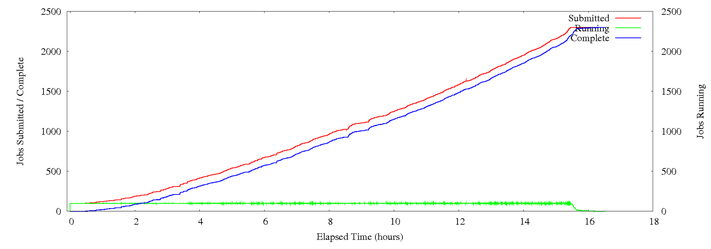


Total CPU time is calculated as the sum of the execution times of each job in the Makeflow.

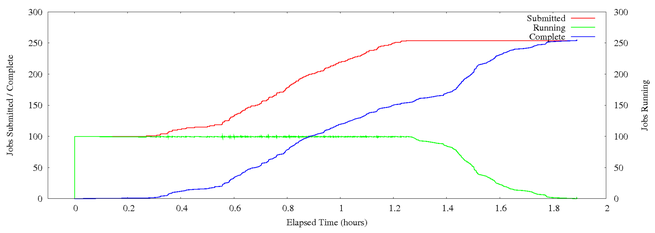


Sorghum bicolor
Utilizing a homogeneous 62-node cluster

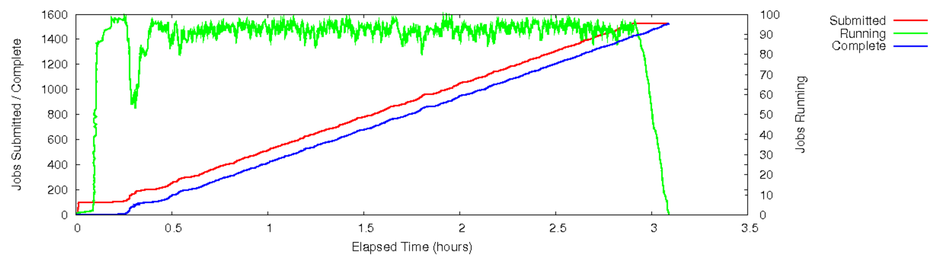
Sorghum bicolor
11.5 million reads (~11 billion bases) aligned to the genome (~738.5 million bases)



Sorghum bicolor
Utilizing 100 processors



Anopheles gambiae M form
2.5 million reads (~1.5 billion bases) aligned to the PEST genome (~273 million bases)



Oryza rufipogon
7 million reads aligned to the *Oryza sativa* genome

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. Journal of Molecular Biology, 215(3), 403 – 410.
Langmead, B., Schatz, M., Lin, J., Pop, M., and Salzberg, S. (2009). Searching for SNPs with cloud computing. Genome Biology, 10(11), R134.
Moretti, C., Olson, M., Emrich, S., and Thain, D. (2009). Highly scalable genomeassembly on campus grids. Many-Task Computing on Grids and Supercomputers (MTAGS).
Ning, Z., Cox, A. J., and Mullikin, J. C. (2001). SSAHA: A Fast Search Method for Large DNA Databases. Genome Research, 11(10), 1725–1729.
Rumble, S. M., Lacroute, P., Dalca, A. V., Fiume, M., Sidow, A., and Brudno, M. (2009). SHRiMP: Accurate mapping of short color-space reads. PLoS Comput Biol,5(5), e1000386.
Thain, D., Tannenbaum, T., and Livny, M. (2002). Condor and the grid. In F. Berman, G. Fox, and T. Hey, editors, Grid Computing: Making the Global Infrastructure a Reality. John Wiley & Sons Inc.
Yu, L., Moretti, C., Emrich, S., Judd, K., and Thain, D. (2009). Harnessing parallelism in multicore clusters with the all-pairs and wavefront abstractions. In HPDC '09: Proceedings of the 18th ACM international symposium on High performance distributed computing, pages 1–10, New York, NY, USA. ACM.

Acknowledgements

We thank Nora Besansky and Philip SanMiguel for providing the genomic data used in this work.

Funding : This work was supported in part by National Science Foundation [grant 0643229] and by the National Institutes of Health [NIAID contract HHSN266200400039C].