BIOINFORMATICS APPLICATIONS NOTE
Vol. 28 no. 1 2012, pages 136–137
doi:10.1093/bioinformatics/btr615

Systems biology
Advance Access publication November 8, 2011

MR-Tandem: parallel X!Tandem using Hadoop MapReduce on Amazon Web Services

Brian Pratt*, J. Jeffry Howbert, Natalie I. Tasman and Erik J. Nilsson
Insilicos LLC, Seattle WA, USA
Associate Editor: John Quackenbush

ABSTRACT

Summary: MR-Tandem adapts the popular X!Tandem peptide search engine to work with Hadoop MapReduce for reliable parallel execution of large searches. MR-Tandem runs on any Hadoop cluster but offers special support for Amazon Web Services for creating inexpensive on-demand Hadoop clusters, enabling search volumes that might not otherwise be feasible with the compute resources a researcher has at hand. MR-Tandem is designed to drop in wherever X!Tandem is already in use and requires no modification to existing X!Tandem parameter files, and only minimal modification to X!Tandem-based workflows.

Availability and implementation: MR-Tandem is implemented as a lightly modified X!Tandem C++ executable and a Python script that drives Hadoop clusters including Amazon Web Services (AWS) Elastic Map Reduce (EMR), using the modified X!Tandem program as a Hadoop Streaming mapper and reducer. The modified X!Tandem C++ source code is Artistic licensed, supports pluggable scoring, and is available as part of the Sashimi project at http://sashimi.svn.sourceforge.net/viewvc/sashimi/trunk/trans_proteomic_pipeline/extern/xtandem/. The MR-Tandem Python script is Apache licensed and available as part of the Insilicos Cloud Army project at http://ica.svn.sourceforge.net/viewvc/ica/trunk/mr-tandem/. Full documentation and a windows installer that configures MR-Tandem, Python and all necessary packages are available at this same URL.

Contact: brian.pratt@insilicos.com

Received on August 30, 2011; revised on October 13, 2011; accepted on November 1, 2011

1 INTRODUCTION

Post-translational modifications (PTMs) of proteins are a valuable source of biological insights, and mass spectrometry is one of the few techniques able to reliably prospect for and identify PTMs. Yet many valuable datasets are not searched for PTMs due to computational constraints on investigators who do not have claim to significant time on a compute cluster. MR-Tandem helps by bringing X!Tandem (Craig and Bevis, 2004) peptide search to cloud computing and using Hadoop to process large datasets in a fast and robust manner.

MR-Tandem is not the first parallel implementation of X!Tandem but it is the first to exploit the scalability and fault tolerance of Hadoop to create large on-demand compute clusters on commodity hardware where MPI implementations may fail.

*To whom correspondence should be addressed.

2 METHODS

2.1 Simultaneous search versus parallel search

There are many solutions for running simultaneous X!Tandem search jobs on a cluster. This is useful and relatively easy to implement, but does not speed up individual search tasks that may possibly take hours or days to complete. Solutions also exist which parallelize the search task itself: standard X!Tandem contains a threading model that allows it to spread the work of an individual search across multiple processor cores, and the X!!Tandem project (Bjornson *et al.*, 2008) extends this model to multiple nodes on a network. Others such as the Parallel Tandem project (Duncan *et al.,* 2005), subdivide searches into smaller independent X!Tandem jobs to be run on network nodes, then synthesize the multiple result sets into a whole.

Unfortunately these multi-node parallel implementations can be quite complicated to implement and maintain. Many, such as X!!Tandem, rely on MPI, a parallel computation framework that is often impractical due to its brittleness as cluster size increases unless specialized network hardware is used to avoid latency problems.

2.2 Hadoop versus MPI

MR-Tandem mimics X!!Tandem in extending X!Tandem's existing threading model out onto a network, but uses Hadoop MapReduce instead of MPI. The principal advantage of Hadoop over MPI is fault tolerance. With MPI a node failure or even a simple network delay will cause an unrecoverable error and the potential loss of hours of computational effort, but Hadoop is designed with these scenarios in mind and any failed work unit is simply retried on a different node. (It is possible to write fault tolerant MPI code, but this is rarely done in practice.) Fault tolerance brings scalability: the increased likelihood of network delays and node failures in a larger cluster can make MPI unusable where Hadoop remains comfortably robust. In practice on AWS we find that it is not possible to reliably establish an MPI ring of more than about 10 commodity (m1.small or m1.large) nodes but we can easily create Hadoop clusters with dozens of such nodes. (AWS does provide 'Cluster Compute' instances that offer better network performance and thus work better with MPI, but these are more expensive to use.).

2.3 MR-Tandem and cloud computing

While MR-Tandem can utilize any Hadoop cluster, special accommodation is made for using AWS EMR to easily create clusters on demand. This has obvious benefits to any researcher in need of extra compute resources without having to negotiate for time on someone else's search cluster.

3 RESULTS

3.1 Installation

MR-Tandem is implemented as a Python script that communicates with AWS EMR or your own Hadoop cluster, and a customized X!Tandem executable that downloads to the cluster from a public

Amazon S3 bucket. The Windows installer program for MR-Tandem ensures the presence of a suitable version of Python and any required libraries, helping install them if needed (Linux and Mac users follow a manual installation process).

3.2 Operation

Where you might have previously invoked a local search as
`tandem my_search_params.xml`
you can now invoke a search in the cloud as
`mr-tandem.py my_aws_info my_search_params.xml`
where `my_aws_info` is a human readable JSON-formatted file containing the information necessary to access your account on AWS. MR-Tandem invokes the Hadoop cluster, transfers any needed data to S3, downloads the MR-Tandem binary to the cluster from a public S3 bucket hosted by Insilicos (alternative download locations can be specified by the user), and starts the search. Any previously transferred data will not be sent again. Results are copied back to your local machine. All file references in the results (protein database, mass spec files) are as they would be if the search had been run locally, making it trivial to plug MR-Tandem into existing systems that use X!Tandem without disrupting downstream tools.

3.3 Scalability and performance compared to MPI

MPI-based X!!Tandem was found on average to run ∼20% faster than MR-Tandem, although we were only able to test this at the low AWS node counts where MPI could be made to work. X!!Tandem's speed advantage here is partly due to MR-Tandem using HDFS disk I/O to pass large (for Hadoop) result sets from the workers. Also, Hadoop guarantees the success of all tasks in one step before proceeding to another, so the job does not proceed to the refinement stage until the slowest search task is complete. X!!Tandem, in contrast, passes search results via ssh, and processes them for refinement as soon as they start to come in, but will fail when any node fails to deliver results. Future work may be able to address both of these limitations.

MR-Tandem scales like other X!Tandem parallel implementations that search against the same proteins on all nodes. Performance initially improves with each added node, but eventually the cost of generating theoretical spectra from the protein database becomes the limiting factor and additional nodes do not improve performance. In tests with 26 172 MS2 spectra (233 MB mzXML file) and 52 415 proteins (33 MB FASTA file) we found MR-Tandem scaled well up to 50 nodes (see Table 1).

Table 1. MR-Tandem scalability

No. of Nodes	1[a]	2	5	10	20	50	100	200
t (hours)	36.12	29.32	11.54	6.69	4.09	1.76	1.28	1.16
Speedup	1	1.23	3.13	5.4	8.83	20.56	28.13	31.21

[a]This is a two-threaded non-Hadoop standard X!Tandem run for baseline comparison.

3.4 Cost

By default MR-Tandem uses five AWS EMR 'm1.small' nodes for Hadoop, at a cost of $0.10 per node per hour. Cluster size and node type can be otherwise specified in the JSON-formatted file that contains the user's AWS credentials. Amazon charges by the hour (1 minute costs the same as 60) so rather than starting a new cluster for each new search, MR-Tandem provides a simple means of serially running multiple parallel computations on a single cluster invocation: the X!Tandem parameter file may be replaced by a text file containing a list of X!Tandem parameter files to be processed in turn. This coarse billing granularity gives rise to some interesting price/performance inflections and the authors hope to do further work on predicting and automating optimal cluster sizes for a given search task, as well as supporting AWS 'Spot Pricing' that allows one to bid for underutilized AWS compute time at much lower rates.

ACKNOWLEDGEMENT

X!Tandem (http://www.thegpm.org/) forms the basis of this work and its core algorithms and functionality are unaltered. The X!!Tandem project (also at thegpm.org) furnished inspiration on ways to extend the X!Tandem threading model onto a network, as well as some object serialization code. The X!Tandem code fork that MR-Tandem builds upon is part of the Sashimi project (http://tools.proteomecenter.org). The Insilicos Cloud Army (http://sourceforge.net/projects/ica/) ensemble machine learning project provided valuable insights on the use of AWS and Hadoop.

Funding: National Institutes of Health (HG006091).

Conflict of Interest: none declared.

REFERENCES

Craig,R. and Beavis,R.C. (2004) TANDEM: matching proteins with mass spectra. *Bioinformatics*, **20**, 1466–1467.

Bjornson,R.D. *et al.* (2008) X!!Tandem, an improved method for running X!Tandem on Collections of Commodity Computers. *J. Proteome Res.*, **7**, 293–299.

Duncan,D.T. *et al*. (2005) Parallel Tandem: a program for parallel processing of tandem mass spectra using PVM or MPI and X!Tandem. *J. Proteome Res.*, **4**, 1842–1847.