

Porting the existing MG-RAST multi-user web application to the cloud

Knowledgebase R&D Pilot Project

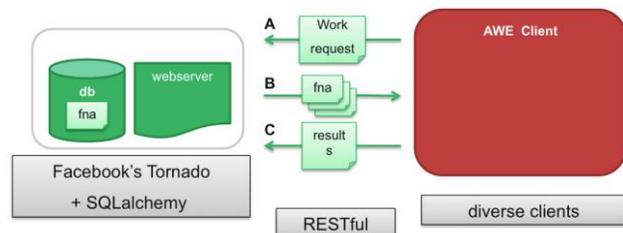
Jared Wilkening, Andreas Wilke, Elizabeth M. Glass, Narayan L. Desai, Folker Meyer
Argonne National Laboratory

The Metagenomics RAST server (MG-RAST) allows public upload and analysis of data via its web portal and has been enabling over 2000 data submitting users since 2007. To continue providing this community resource to groups in over 30 countries the team supporting the server had to identify a means of scaling up analytical capacity. The task seemed daunting as for the analysis of 1 GB of metagenomic data, MG-RAST requires approx. 2,000 CPU hours on a recent Intel Nehalem machine with moderate memory (8), the computational cost dominated by computing sequence similarity searches. For this project we investigated computations in the MG-RAST computational pipeline to determine their suitability to the cloud computational paradigm.

The MG-RAST system provides a web portal and a computational pipeline for the analysis of metagenomics datasets. Users submit sequence data sets via the MG-RAST portal, a web interface implemented as a series of Perl cgi scripts. This data is loaded into a Postgres database, along with information about which analytics should be performed and metadata about the dataset. This data is used, in turn, to submit jobs into the computational pipeline, AWE, which manages work execution.

Due to a combination of increasing demand for analysis via MG-RAST, as well as the quickly increasing data set sizes, it has become clear that dedicated computing resources will not provide sufficient capacity for even the short term, with considerably larger shortfalls in the medium term. Our first approach to tackling this problem was to scale out the MG-RAST backend to existing shared resources at Argonne, followed by an approach using new cloud resources. This change posed some technical challenges, due to differences in the general infrastructure provided on shared systems, as well as the switch to using distributed resources.

The similarity analysis stage of the MG-RAST system was our initial target. This stage is implemented using NCBI BLAST. It is a good candidate for distribution, both because of its large overall resource consumption, and the computation uses a large fixed database that changes infrequently. The only input to this stage is a small query sequence, easily transmitted from AWE. AWE consists of a centralized set of python daemons that can communicate with clients via a RESTful interface. This approach is widely portable, as clients need only be able to perform HTTP requests to the server. The work queue, as well as results and statistics, are stored in a Postgres database. We have used Facebook's Tornado framework to build a lightweight and efficient set of front tier web servers.



After becoming accustomed to dealing with performance issues, we were able to retune our database, modify our work management reliability scheme, and scale a single backend up to 500 compute nodes without issue. Further optimizations can be applied when they are needed; the architecture is designed to scale among multiple backend machines. However, since our current approach has been sufficiently scalable for the compute resources we have access to, we have not pursued this issue.