Astronomical Data Analysis Software and Systems XXI
ASP Conference Series, Vol. 461
Pascal Ballester, Daniel Egret, and Nuria P. F. Lorente, eds.

High-Performance Compute Infrastructure in Astronomy: 2020 Is Only Months Away

Bruce Berriman,[1] Ewa Deelman,[2] Gideon Juve,[2] Mats Rynge,[2]
and Jens S. Vöckler[2]

[1]*Infrared Processing and Analysis Center, California Institute of Technology, Pasadena, CA 91125, USA*

[2]*Information Sciences Institute, University of Southern California, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292, USA*

Abstract. By 2020, astronomy will be awash with as much as 60 PB of public data. Full scientific exploitation of such massive volumes of data will require high-performance computing on server farms co-located with the data. Development of this computing model will be a community-wide enterprise that has profound cultural and technical implications. Astronomers must be prepared to develop environment-agnostic applications that support parallel processing. The community must investigate the applicability and cost-benefit of emerging technologies such as cloud computing to astronomy, and must engage the Computer Science community to develop science-driven cyberinfrastructure such as workflow schedulers and optimizers.

We report here the results of collaborations between a science center, IPAC, and a Computer Science research institute, ISI. These collaborations may be considered pathfinders in developing a high-performance compute infrastructure in astronomy. These collaborations investigated two exemplar large-scale science-driver workflow applications: 1) Calculation of an infrared atlas of the Galactic Plane at 18 different wavelengths by placing data from multiple surveys on a common plate scale and co-registering all the pixels; 2) Calculation of an atlas of periodicities present in the public Kepler data sets, which currently contain 380,000 light curves. These products have been generated with two workflow applications, written in C for performance and designed to support parallel processing on multiple environments and platforms, but with different compute resource needs: the Montage image mosaic engine is I/O-bound, and the NASA Star and Exoplanet Database periodogram code is CPU-bound. Our presentation will report cost and performance metrics and lessons-learned for continuing development.

Applicability of Cloud Computing: Commercial Cloud providers generally charge for all operations, including processing, transfer of input and output data, and for storage of data, and so the costs of running applications vary widely according to how they use resources. The cloud is well suited to processing CPU-bound (and memory bound) workflows such as the periodogram code, given the relatively low cost of processing in comparison with I/O operations. I/O-bound applications such as Montage perform best on high-performance clusters with fast networks and parallel file-systems.

Science-driven Cyberinfrastructure: Montage has been widely used as a driver application to develop workflow management services, such as task scheduling in distributed environments, designing fault tolerance techniques for job schedulers, and developing workflow orchestration techniques.

Running Parallel Applications Across Distributed Cloud Environments: Data processing will eventually take place in parallel distributed across cyber infrastructure environments having different architectures. We have used the Pegasus Work Management

System (WMS) to successfully run applications across three very different environments: TeraGrid, OSG (Open Science Grid), and FutureGrid. Provisioning resources across different grids and clouds (also referred to as Sky Computing), involves establishing a distributed environment, where issues of, e.g, remote job submission, data management, and security need to be addressed. This environment also requires building virtual machine images that can run in different environments. Usually, each cloud provides basic images that can be customized with additional software and services. In most of our work, we provisioned compute resources using a custom application, called Wrangler. Pegasus WMS abstracts the architectures of the compute environments away from the end-user, and can be considered a first-generation tool suitable for scientists to run their applications on disparate environments.

1. Surviving The The Data Tsunami: Cooperation Between Astronomers and Computer Scientists

The astronomical community is identifying the practices needed by astronomers and computer scientists working in cooperation to manage and process the vast quantities of data (Berriman & Groom 2011) that are expected to become publicly available by 2020. By then, distributed, parallel processing of these data sets will be routine. Astronomers at the Infrared Processing and Analysis Center (IPAC) and computer scientists at the USC Information Sciences Institute have for the past decade collaborated on understanding and developing cyber infrastructure tools can best support the needs of scalable astronomical workflow applications in creating new scientific products that meet stringent scientific specifications. This paper describes how the collaboration has realized some of the aforementioned recommended practices, including the development of environment-agnostic code, user-friendly tools for managing and processing data in distributed environments, rigorous evaluation of the applicability of emerging technologies, and cyber infrastructure developed to meet modern scientific needs (Berriman & Groom 2011).

2. Environment-Agnostic Computing and Workflow Management Tools

Montage is an example of an application *designed* to run on multiple, distributed environments. It is written in ANSI-C, has no third-party dependences other than common astronomy libraries. It runs on essentially all *nix platforms. It consists of standalone modules that perform the tasks needed to calculate mosaics. The modules are plugged together and controlled by an executive library. While Montage supports MPI, we have invariably run Montage with the Pegasus Workflow Management System manager (Deelman et al. 2005). It maps an abstract workflow to a form that can be executed on a cluster, cloud, or grid. It allows scientists to run complex parallel workflows without knowledge of underlying resources. At first, Pegasus was used to support Montage's contractual obligation to meet performance goals running on a grid. These goals led to the use of job clustering techniques to increase the computational granularity of workflow jobs. This optimization led to performance figures better than those obtained by the MPI version of Montage. The availability of, on the one hand, environment-neutral applications such as Montage, and workflow management tools that can support multiple

compute platforms, positioned the collaboration to investigate new compute technologies and support the development of science-driven computing infrastructure.

3. Scientific Applicability of Cloud Computing

Commercial clouds represent a new way of purchasing processing power and storage. A comparative study (Juve et al. 2009) of the cost and performance of applications with widely differing resource usage, running on the cloud and on a high-performance cluster (equipped with a high-speed network and a parallel file system) showed that users should perform a cost-benefit analysis to identify the most cost-effective processing and data storage strategy on the cloud. The Amazon Cloud (EC2) offers a good value for compute- and memory-bound applications, while parallel file systems and high-speed networks offer the best performance for I/O-bound applications such as Montage. Overall, the cloud may be best suited to one-time bulk-processing ("high burst") tasks, providing excess capacity under load, and running test-beds.

4. Science-driven Cyber Infrastructure

The computer science community have used Montage, due to its data availability, open source codes, ease of use, and scalability, as an exemplar application on grid, cluster and cloud platforms in developing the next generation of data-aware cyber-infrastructure, including (see Berriman et al. (2011) for references): Task scheduling in distributed environments; designing job schedulers for the grid; designing fault tolerance techniques for job schedulers; exploring issues of data provenance in scientific workflows; exploring the cost and performance of scientific applications running on clouds; developing high-performance workflow restructuring techniques; developing application performance frameworks, and developing workflow orchestration techniques.

5. A Multiwavelength Infrared Image Atlas of the Plane of the Galaxy

An example of an I/O bound workflow is the calculation of an infrared atlas of the Galactic Plane at 18 different wavelengths. This is a workflow of workflows, with each wavelength consisting of 900 Montage runs. The Montage engine performs all the tasks needed to assemble a set of input images into a mosaic: processing the input images to the required spatial scale, coordinate system, image projection; rectifying the background emission across the images to a common level, and co-adding the processed, rectified images to make the final output mosaic. The result will be a multiwavelength image atlas of the galactic plane that appears to have been measured with a single instrument observing 18 wavelengths. When running these computations, the bottleneck is not the available cores, but filesystem quotas and I/O rates. Each Montage run in this case takes on average 30 hours, but can vary significantly depending on available I/O, both from the archive containing the source images, and the filesystem tied to the computational resource. To make sure that the computation is not exceeding disk quotas, the workflow is usually limited to only release a relatively small amount of work at any given time. In order to not overwhelm the archive site, a caching system with a rate limiter against the archive site is used.

6. Atlas of Periodicities in the Time-Series Data Sets Released by the Kepler Satellite

An example of a "high burst" use-case is to generate an atlas of periodicities present in the public time series data sets released by the NASA Kepler mission. The computations use the periodogram application developed by the former NASA Star and Exoplanet Database (NStED). As with Montage, the code is designed for portability and scalability. It is written in C, and is easy to parallelize because each frequency can be processed independently of all other frequencies. The application spends 90% of its time processing data and is therefore strongly CPU-bound. A trial product composed of periodograms of 210,000 light curves resulted in a cost of $300 on the Amazon EC2 cloud. Using 128 processors, the calculations took 26 hours with the classical Lomb-Scargle algorithm (Berriman et al. 2010). Costs do, however, grow rapidly with the number of curves and the density of frequencies sampled, and so we have begun to investigate the use of academic clouds for these calculations. We have processed a subset of 33,000 Kepler light curves on eight processors provisioned on the Magellan Cloud `http://www.nersc.gov/nusers/systems/magellan/` and the FutureGrid platforms `http://futuregrid.org`. The performance is comparable with that on Amazon EC2 (Vöckler et al. 2011). What remains to be seen is whether the level of service provided by academia can be on the par with that delivered by commercial providers. Finally, we have successfully distributed this processing across two very different environments, the TeraGrid and the Open Science Grid. The experiment took advantage of the GlideinWMS and Corral Frontend technologies (Rynge et al. 2011).

Acknowledgments. G. B. Berriman is supported by the NASA Exoplanet Science Institute at the Infrared Processing and Analysis Center, operated by the California Institute of Technology in coordination with the Jet Propulsion Laboratory (JPL). USC/ISI researchers are funded by the National Science Foundation under Cooperative Agreement OCI-0438712 and grant CCF-0725332.

References

Berriman, G. B., Good, J., Deelman, E., & Alexov, A. 2011, Philos Transact A Math Phys Eng Sci, 369, 3384

Berriman, G. B., & Groom, S. L. 2011, Queue, 9, 21:20. URL `http://doi.acm.org/10.1145/2039359.2047483`

Berriman, G. B., Juve, G., Deelman, E., Regelson, M., & Plavchan, P. 2010, in Proceedings of the 2010 e-Science in Astronomy Conference, Brisbane, Australia. `arXiv:1010.4813`

Deelman, E., et al. 2005, Scientific Programming Journal, 13, 219

Juve, G., Deelman, E., Vahi, K., Mehta, G., Berriman, B., Berman, B. P., & Maechling, P. 2009, in Workshop on Cloud-based Services and Applications in conjunction with 5th IEEE International Conference on e-Science (e-Science 2009)

Rynge, M., et al. 2011, in Proceedings of the 7th IEEE International Conference on e-Science (e-Science 2011)

Vöckler, J.-S., Juve, G., Deelman, E., Rynge, M., & Berriman, B. 2011, in Proceedings of the 2nd international workshop on Scientific cloud computing (New York, NY, USA: ACM), ScienceCloud '11, 15. URL `http://doi.acm.org/10.1145/1996109.1996114`