WS-VLAM as an Environment for Data Intensive Applications

A.S.Z. Belloum[1], R. Cushing[1], S. Koulouzis[1], M. Baranowski[1], and M.T. Bubak[1,2]

[1] University of Amsterdam, Institute for Informatics, Science Park 904, 1098 XH Amsterdam, NL
[2] AGH University of Science and Technology, Department of Computer Science, Kraków, PL

Keywords: e-Science, data intensive science, workflow systems, e-infrastructures

1. Introduction

As data forms the core subject of research, all processes related to it such as authoring, publishing, managing, sharing, accessing, reusing and annotating dramatically influence the efficiency of scientific investigations. We have developed a workflow management system that supports both domain scientists and workflow developers enabling to investigate optimal development and execution of compute and data intensive scientific applications on available e-infrastructures [1, 2]. WS-WLAM is flexible and easily extendable.

2. Description of WS-VLAM

WS-VLAM covers the entire lifecycle of scientific workflows from design through execution phase to sharing and reuse complete workflows and their components[1]. The central system module is a message exchange server that provides communication between workflow tasks, coordination, and provenance capture. A top level scheduler (Enactment Engine) is responsible for orchestrating workflows, and a bottom level scheduler (Resource Submission Scheduler) schedules tasks onto resources through matchmaking (Fig. 1). The bottom level scheduler can manage multiple resources such a grids, clouds, and clusters [3].

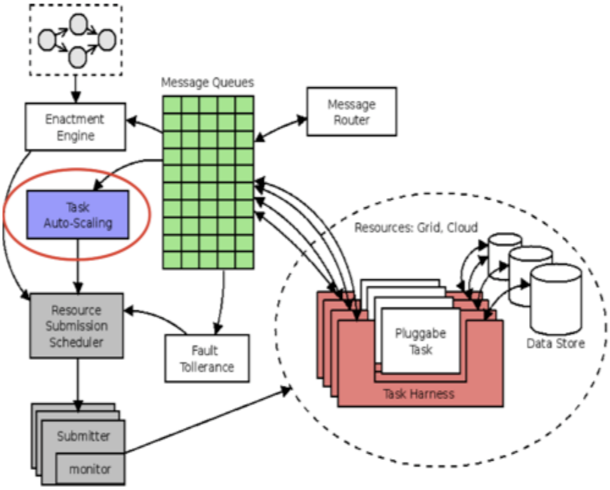

Fig. 1. Loosely coupled WS-VLAM engine architecture components revolving around a core messaging component.

[1] www.science.uva.nl/~gvlam/wsvlam

Results Monitoring is performed at the workflow level and workflow component levels. At the workflow level, the end user can follow the state of a workflow submission and check whether the workflow is pending, submitted, running, or completed [1]. WS-VLAM provides a data-aware web service to efficiently transport large datasets between web services. ProxyWS component coordinates the transportation of large data volumes from remote resources (secure FTP, GridFTP), supports data transfers and data streams between web services as well as enables legacy web services to reference data that would be delivered via SOAP; web services can process and deliver larger datasets with streaming [2].

3. Sample results

Farming and scaling capabilities are achieved through a dedicated module on top the message queue. Since workflow tasks communicate over the message exchange, the latter can be used to inspect the state of the workflow and deduce which tasks are overloaded by looking at their respective data queue sizes [3]. The concept of workflow as a service (WfaaS) has been elaborated to increase the performance and minimize the overheads of workflow farming. Two approaches to workflow farming are implemented: task-level where task harness acts as services by being invoked on which task to load, and data-level where the actual task is invoked as a service with different chunks of data to process [4, 5].

4. Conclusions and future work

Prediction-based auto-scaling can be applied to data-centric workflows as a way to accelerate data processing rates within scientific workflows. The ability of scaling tasks independently enables replication of tasks to match the data production rate. This minimizes work bottlenecks and reduces total makespan. Through task harnessing we showed how scientific logic could be separated from underlying communication and data transport intricacies. Another area of interest is the possibility for peer-to-peer web service communication. There is possibility for a web service to detect that services are running on the same network and then open direct socket connections between them.

Acknowledgments. This work was partially funded by COMMIT P20 WP8 and WP9 (`http://www.commit-nl.nl`) as well as VPH-Share (`http://www.vph-share.eu`) project.

References

1. A. Belloum, M. A. Inda, D. Vasunin, V. Korkhov, Z. Zhao, H. Rauwerda, T. M. Breit, M. Bubak, L. O. Hertzberger: Collaborative e-Science Experiments and Scientific Workflows. IEEE Internet Computing 15(4) 39-47 2011, DOI 10.1109/MIC.2011.87
2. S. Koulouzis, R. Cushing, K. A. Karasavvas, A. Belloum, M. Bubak: Enabling Web Services to Consume and Produce Large Datasets, IEEE Internet Computing 16 (1) 52 - 60 2012, DOI 10.1109/MIC.2011.138
3. R. Cushing, S. Koulouzis, A. Belloum, M. Bubak, Prediction-based Auto-scaling of Scientific Workflows, MGC '11 Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science, December 12, 2011, Lisbon, Portugal, DOI 10.1145/2089002.2089003
4. R. Cushing, A. Belloum, V. Korkhov, D. Vasyunin, C. Leguy, M. Bubak: Workflow as a Service: An Approach to Workflow Farming, ECMLS'12 - The Third International Emerging Computational Methods for the Life Sciences Workshop, June 18, 2012, Delft, The Netherlands (to be incorporated into the ACM Digital Library)
5. Reginald Cushing, Spiros Koulouzis, Adam Belloum, Marian Bubak: Applying Workflow as a Service Paradigm to Application Farming submitted to Concurrency and Computation: Practice and Experience.