ServMark: A Framework for Testing Grid Services

C. Dumitrescu[⋆,†], A. Iosup[⋆], H. Mohamed[⋆], Dick H.J. Epema[⋆], M. Ripeanu[∘], N. Tapus[+], I. Raicu, I. Foster[∗]

Contact: {`servmark-announce`, `servmark-user`, `servmark-dev`}`@globus.org`

[⋆]*Delft University of Technology, NL,* [†]*University of Muenster, DE,* [∘]*University of British Columbia, CA,*
[+]*Politehnica University of Bucharest, RO,* [∗]*University of Chicago, Illinois, US*

I. Introduction

The dynamicity, the heterogeneity, or simply the sheer scale of today's grids expose in grid services problems of performance, functionality, and reliability. While testing under realistic conditions is a proven industrial method, little has been done in grids in this direction. Early testing in real grids reveals failure rates from 10% and up to 45% [1], [2], [3], and functionality problems of around 1 every 3 tests for widely-installed grid services [4][1]. To further the development of grid services testing in large-scale settings, we present in this work the SERVMARK framework.

In SERVMARK, we address two orthogonal research questions : (1) *How to test a large-scale, distributed, and (grid-)service-based environment?* and (2) *How to generate realistic testing traces for a wide-range of testing scenarios?*. To the best of our knowledge, and as the main contribution of this work, ours is the first approach that answers both questions in the context of functionality testing and system tuning, of performance testing, and of reliability testing.

II. The ServMark Framework

In SERVMARK, users have one of two roles: (1) The *Test Manager* defines and configures the test. In particular, the Test Manager specifies the test workload models, and how to mix them if several are specified. The model parameters can be directly specified or automatically extracted from existing grid workload data (i.e., grid traces). (2) The *Test User* does post-test results analysis.

The SERVMARK framework includes five components (see Figure 1): (1) The *Controller* coordinates the testing procedure. (2) The *Workload Modeler* generates the model parameters for the indirect parameter specification. (3) The *Centralizer* receives workload and other test specifications from the Controller, coordinates a group of Testers (described below) for distributed testing, and submits (summarized) results to the Data Warehouse (described below). (4) The *Tester* generates and submits workloads based on the specification from the Centralizer, and submits (summarized) results back to the Centralizer. (5) The *Data Warehouse* stores test data, and performs simple data mining and analysis. The framework allows for additional data mining and analysis mechanisms to be added.

To implement the SERVMARK, we leverage two existing grid testing tools: the DiPerF distributed grid testing tool [6], and the GrenchMark grid workload generator and submitter [1].

[1]The failure rate in grids is much higher than that of contemporary large-scale parallel production installations [5].

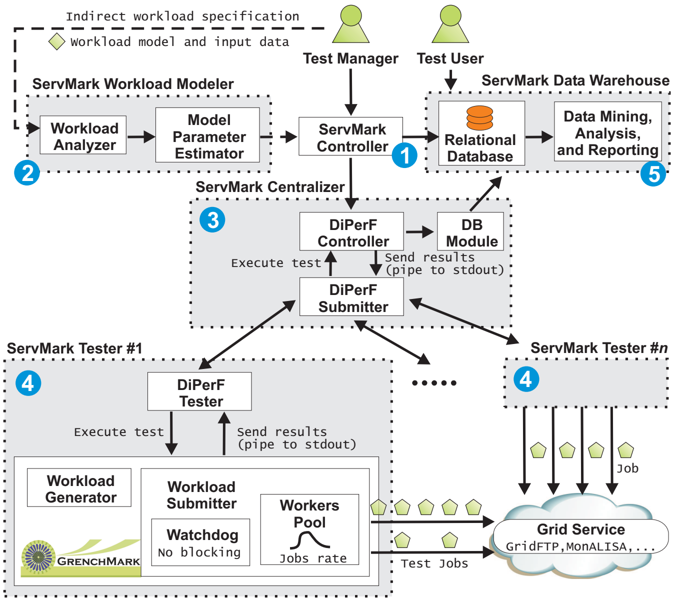

Fig. 1. The ServMark framework design and implementation.

III. Research and Technical Directions

More research results are needed at least in the following areas: (1) *Fault-tolerant testing*, such that grid testing can be provided as a transparent service to the user; (2) *Reproducible system conditions*, such that at least the background load is consistently reproduced; and (3) *The design and validation of grid workload models*, such that the generated wrokloads are truly realistic. We plan to address them in the context of SERVMARK.

The SERVMARK implementation has been already shown to work in the context of testing simple grid services, and is currently used to test the GridFTP grid service. For the future, we also plan to develop distributed versions of the Controller and Centralizer components.

Availability

ServMark is available as a Globus Incubator project at: `http://dev.globus.org/wiki/Incubator/ServMark`

References

[1] A. Iosup and D. H. J. Epema, "Grenchmark: A framework for analyzing, testing, and comparing grids." in *CCGRID*.
[2] O. Khalili, Jiahue He, et al., "Measuring the performance and reliability of production computational grids," in *GRID*, 2006.
[3] H. Li, D. Groep, L. Wolters, and J. Templon, "Job failure analysis and its implications in a large-scale production grid." in *e-Science*, 2006.
[4] A. Iosup, D. Epema, P. Couvares, A. Karp, and M. Livny, "Build-and-test workloads for grid middleware: Problem, analysis, and applications," in *CCGRID*.
[5] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," in *DSN*, 2006.
[6] C. Dumitrescu, I. Raicu, M. Ripeanu, and I. T. Foster, "Diperf: An automated distributed performance testing framework." in *GRID*, 2004.