Performance Cockpit: Systematic Measurements and Analyses

Dennis Westermann
SAP Research
Vincenz-Priessnitz-Str. 1
Karlsruhe, Germany
dennis.westermann@sap.com

Jens Happe
SAP Research
Vincenz-Priessnitz-Str. 1
Karlsruhe, Germany
jens.happe@sap.com

ABSTRACT

Measurement-based performance evaluations are heavily used in practice to test system behavior under load, identify resource bottlenecks, or size system landscapes. Existing literature provides guidelines on how to conduct performance evaluations correctly. Many tools (e.g. for load generation, monitoring, or statistical analyses) provide basic assets to conduct such evaluations. However, the wide range of knowledge required to conduct performance evaluations and control the available tools restricts the group of users to a small set of performance experts. Additionally, the large effort to set up systems for performance evaluations often limits their application. In this demo paper, we present a framework that encapsulates best practices and allows for separation of concerns regarding the different aspects of a performance evaluation. The Performance Cockpit provides a single point of configuration for performance analysts and orchestrates plug-ins provided by corresponding experts. The resulting flexibility and automation enables new approaches for quality assurance and lowers the hurdles for conducting performance evaluations.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques

General Terms

Performance, Measurement

1. INTRODUCTION

Evaluating the performance (response time, throughput, resource usage) of software systems is an important task during its whole life cycle (e.g. design, development, provisioning, use, or upgrade). Performance evaluations help to plan capacities, evaluate and analyse software architectures, or identify performance bottlenecks.

Although, there are common rules and best practices on how to conduct performance evaluations correctly (e.g. [2]), the performance measures, the measurement and monitoring tools, as well as the analysis techniques are very situation specific. Due to the heterogeneity of systems a variety of tools and approaches exist for generating load, monitoring the systems behavior, and analysing measurement results. Furthermore, the goal of an evaluation depends on the stage in the life cycle and the specific scenario that is under study. For some systems standard application benchmarks (such as provided by SPEC) can answer specific performance questions. For customized performance evaluations tools for load generation, monitoring and analysis must be appropriately configured and controlled. Moreover, today software systems continuously evolve which requires repeating evaluations. In summary, conducting a performance evaluation requires a wide range of knowledge with respect to best practices, the system under test, load generation, monitoring, and analysis.

The Performance Cockpit, which we present in this demo paper, simplifies performance evaluations by separation of concerns. Experts of the system under test, tools, or benchmarks create plug-ins that allow to control these applications via a common interface in a central tool. The framework orchestrates the different plug-ins following a chosen measurement strategy and implementing best practices of performance measurements (e.g. ensure that enough data is sampled to derive statistically significant statements). The plug-in based architecture allows reuse of existing components and eases the repetition of experiments in similar settings. Performance analysts use the Performance Cockpit by selecting required plug-ins, specifying the parameters that are to be studied, and defining the goal of the measurements. In [4], we presented the architecture and the idea of our approach. In our demo, we present the first stable version of the Performance Cockpit. We applied this version to conduct measurements for our ongoing research and for quality assurance in product development. In the following, we give a brief overview of the approach and outline the benefits for different target groups. We conclude the paper by a discussion of factors critical to success and a description of our next steps.

2. APPROACH

The major design principles of our approach are separation of concern and abstraction. This allows us to offer a single application that implements best practices of software performance evaluation but is highly extendable. The framework hides the complexity of benchmarks, system administration, tool configuration, etc. from the performance analyst by providing a single point of configuration. Figure 1 depicts the main blocks of the framework as well as the roles related to them.

The *User Control* block is the entry point for the performance analyst where he can specify the measurements. This includes defining the parameters that should be varied,

ICPE'11, March 14–16, 2011, Karlsruhe, Germany.
ACM 978-1-4503-0519-8/11/03.

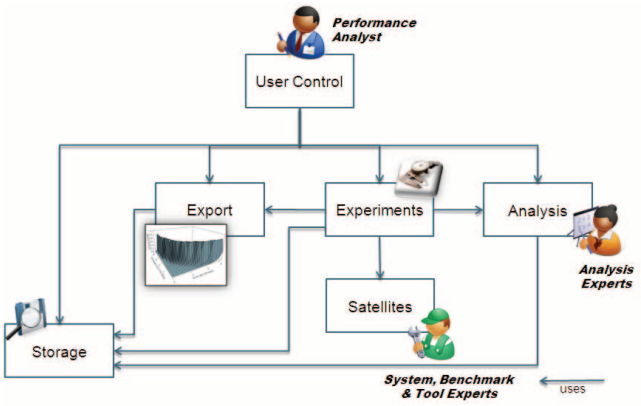

Figure 1: The Performance Cockpit Approach

defining the parameters that should be observed, selecting a measurement strategy, selecting an analysis technique, selecting an export mechanism, and finally starting the measurements. In order to provide such a simple interface to the performance analyst, the other roles have to create appropriate plug-ins. The analysis experts provide plug-ins to run different statistical analyses. Experts of software components, monitoring tools, benchmarks provide the corresponding plug-ins to control the software. These plug-ins run on different *Satellites*. A *Satellite* is a machine that is either part of the system under test or part of the load generating/monitoring. The *Experiments* block forms the core of the framework. The measurements are executed by orchestrating the different plug-ins. The challenges we have to cope here are described in [3]. The measurement data provided by the different satellites is persisted in the *Storage* part of the framework. This is again a common interface where for instance different kinds of databases can be plugged in. Finally, the *Export* block includes different plug-ins that allow to export the measurement data as well as the results of the statistical analyses. There are various possible kinds of exports both the rather simple ones like CSV or plots and more complex ones such as a queuing petri nets or sizing guidelines.

3. BENEFITS

In this section, we describe the potential and the benefits that we see for different target groups when using or contributing to the Performance Cockpit.

IT Companies

In an IT company, the separation of concerns and the resulting ease of use when focusing on the configuration only can lower the hurdle of executing performance evaluations. This supports the adherence to product standards and thus increases the product quality. Additionally, the concept of reusable plug-ins for performance evaluations can save costs and can avoid frequently occurring mistakes. Finally, the efficient measurement strategies provided in the cockpit and the automation of measurements and analyses can bring up novel approaches for process and product improvements in IT companies.

Performance Engineering Research Community

Researchers in the performance engineering community frequently conduct measurements and analyses. Examples are case studies for their work, resource demand estimations for a modeling approach or running benchmarks to demonstrate scalability of a developed system. There are already common scenarios that are used by a wide range of researchers e.g. the SPEC benchmarks, CoCoME or the Dell DVD Store. However, controlling and analyzing these scenarios is done by each researcher every time anew although it is often the same procedure. If the plug-ins to control these scenarios (or any other kind of application) as well as plug-ins for analyses and exports would be available as part of an open source project, researchers could benefit from the work of others and save a lot of time when conducting measurements and thus focus on their actual research. So, the Performance Cockpit can provide a platform for interested researchers to cooperate and share their work.

Open Source Community

Many open source projects do not have the resources to run extensive performance tests. Performance engineering researchers in turn are often interested in validating their approaches on real applications. However, conducting a case study on a large open source application causes high training effort. Open source projects can provide the plug-ins to control their software and enable performance engineers to use these plug-ins to run their case study with the Performance Cockpit and report the results back to the open source project. Thus, there could be a win-win situation between performance engineers and open source projects.

4. CONCLUSIONS

The Performance Cockpit is an approach to ease the execution of measurements by automatically orchestrating different plug-ins which solve a certain problem of a performance evaluation. If the necessary plug-ins are available a performance analyst simply specifies the desired measurement scenario and the Performance Cockpit then runs these measurements automatically. The framework is not a tool for instrumentation, load generation, monitoring or statistical analyses. Instead it targets the automated orchestration of available software. Critical factors of success for such a framework are for instance the ease-of-use, well-documented plug-ins, high quality plug-ins for analyses, or a critical mass of contributors. We use the Performance Cockpit already for our ongoing research in collaboration with other research institutes [1]. Moreover, we are going to publish the first stable version as an open source project.

5. REFERENCES

[1] J. Happe, D. Westermann, K. Sachs, and L. Kapova. Statistical inference of software performance models for parametric performance completions. In *Proceedings of QoSA 2010*, LNCS. Springer-Verlag, 2010.

[2] R. Jain. *The art of computer systems performance analysis*. Wiley Interscience, New York, 1991.

[3] D. Westermann and J. Happe. Towards performance prediction of large enterprise applications based on systematic measurements. In *Proceedings of WCOP 2010*, Karlsruhe, Germany, June 2010.

[4] D. Westermann, J. Happe, M. Hauck, and C. Heupel. The performance cockpit approach: A framework for systematic performance evaluations. In *Proceedings of 36th EUROMICRO SEAA 2010*. IEEE Computer Society, 2010.