Semana de Engenharia 2010
Guimarães, 11 a 15 de Outubro

Self-Managing Service Platform

Nuno A. Carvalho and José Pereira
CCTC - Departamento de Informática
Universidade do Minho
Braga, Portugal
Email: nuno,jop@di.uminho.pt

KEYWORDS

Dependability, Scalability, Resilience, Self-management

MOTIVATION

The increasing complexity and size of a typical distributed system makes it very hard to understand the role of each of its components, such as middleware layers and physical resources, and the impact of their interactions in overall system performance. This task is even harder when systems span several administrative and geographical domains, when a diversity of applications are hosted within a single virtualized infrastructure, or when the sheer number of components leads to subtle emergent properties.

Consider the following example. On one hand, there is a growing acceptance of abstractions for concurrent programming such as software transactional memory in the FenixEDU system or Erlang-style actors with immutable messages. These have however an impact in memory usage and garbage collection overhead. On the other hand, memory management overhead has a profound impact in group communication protocols, which exhibit degraded throughput and arbitrary partitioning. The combination of both might have thus a dramatic effect: when the system is performing at peak throughput, and thus most useful, it becomes unstable.

Consider also a second example. Service platforms in telecom operators are structured as very high throughput multi-tier servers, handling diverse functions such as subscriber profiles and billing. These rest on a combination of application modules and a database system, that is increasingly dynamic and complex as it converges with common middleware and keeps up with the demand for novel services. The consequence of such lack of a deep understanding of performance and availability factors is the need for over-provisioning at multiple levels to ensure desired throughput during peak periods, such as Christmas Eve.

The net result is an increase of systems management and infrastructure costs. Coping with this has lead to focus on autonomic systems and on achieving self-* properties. The goal is that the system components regulate in a decentralized fashion to improve performance, resilience, and efficiency. In detail, many existing proposals rest on feedback control loops to adapt system parameters or structure. Adaptation policy is defined explicitly by administrators or implicitly derived from a model of the system. In either case, the result is only as good as the underlying understanding of the system as a whole.

PROBLEM STATEMENT

It is difficult to model the behavior of large distributed systems, both in terms of number of components, as in their reactions to high loads. This is because (1) the performance of the system depends on how its composition is done as much as on its individual components; and (2) the behavior of each component under load is not easily evaluated during development. This lack of understanding leads to incorrect component models, with missing dependencies and interactions, that severely limit the applicability of autonomic management methods.

APPROACH

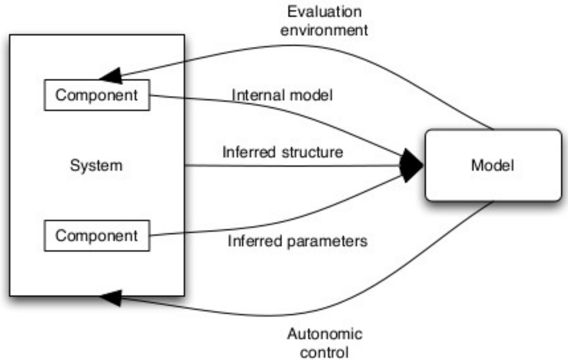

Figure 1: Overview of the approach.

The proposed approach is to seek the virtuous cycle of Figure 1 in which a model is used to improve the evaluation and understanding of individual components and to improve autonomic control, while at the same time, the model itself is improved by deeper insight on individual components as well as actual data gathered from live systems. The result should be to iteratively improve the effect of both manual and self-management mechanisms.

An example of the first direction is the usage of resource models, such as the garbage collector, to improve evaluation of middleware components. Another example is taking advantage of the estimates to make adjustments to the system configuration, particularly in terms of resources allocation, but mainly through discovery of the limits of the system and also to trigger alarms upon abnormal events, which require human intervention, such as the estimates of throughput outside of normal ranges.

An example of second direction is to derive the architecture of a composite system from OSGi bundles used to deploy it, as used in the Serpentine system. This might require enriching it with information on its dynamics, which is then used to derive estimates of their behavior. This could be further improved by inferring the dynamics based on profiling results.

OBJECTIVES AND RESULTS

This works aims to improve understanding and configuration of nowadays distributed systems, through the automation of these two tasks.
As a result, we want to get an improved autonomic element, capable of self learning, forecast and manage the system.

IMPACT

With this solution we want to strengthen the confidence that the owners and administrators have on their systems, because they will always be aware of the system state and behaviour, and most important is limits and correctness. Thus allowing to devote the time previously dedicated to the system management on addressing new challenges.

On the other hand, will also improve the development and evaluation cycle, reducing the time needed between interactions, as their assessment will be continuous, but especially will enable the unequivocal identification of problems that arise.

AUTHOR BIOGRAPHIES

NUNO A. CARVALHO went to the Universidade do Minho, where he studied Computer Science and Systems Engineering technology and obtained his degree. He worked as a researcher in the IST FP6 project ”GORDA - Open Replication of Databases” and P-SON project “Probabilistically-Structured Overlay Networks”. Currently, he is in the MAP-i Doctoral Programme doing his PhD entitled “Self-Managing Service Platform”. His e-mail address is: nuno@di.uminho.pt and his web page can be found at http://gsd.di.uminho.pt/members/nac.