INFRASTRUCTURE FOR MODEL-BASED CONTROL OF DISTRIBUTED IT SYSTEMS

Gergely János PALJAK
Advisors: András PATARICZA, Tamás KOVÁCSHÁZY

I. Introduction

Modern system management applies a feedback control loop scheme (Fig. 1) for guaranteeing a high level of service by (re-)allocating redundant resources in the system to critical functions.

Such feedback control in autonomic computing continuously monitors the service level, for instance performance and availability, and upon an unacceptable deviance triggers health maintenance reconfiguration actions according to a predefined control policy [1, 2]. While traditional heuristic design methodologies were proven extremely useful in server configuration composed of a few, or a few of tens of servers, recent trends tend to create clouds composed of several millions of computing nodes. The complexity of such large-scale infrastructures prohibits the further use of traditional heuristic design methods, especially due to the extreme number of state variables, complex, stochastic, and non-linear interactions. In addition, the characteristics of typical infrastructures and applications are not explored yet deeply; accordingly, the collection of experimental data and checking the candidate control policies in an experimental environment is of an outmost importance.

My objective is the composition of a general purpose environment for data acquisition and experimental control policy development by creating a performance and availability control framework around Matlab, the leading-edge system identification and control implementation software. The measurement environment is based on standard system monitoring tools and a fault-injection engine.

II. Infrastructure for monitoring and control

The *control framework* collects data from a set of *sensors* provided by platform- and application-specific measurement agents in system monitoring. This intelligent *decision-making* unit instructs system provisioning (as *actuator*) to meet or approximate optimization goals (set points). A pilot application infrastructure (Fig. 2) emulates a scaled-down datacenter, over which the framework measures and processes software and platform performance metrics in realistic scenarios. This pilot system adheres to widely-accepted standards and best practices, and is reconfigurable in runtime.

Nagios, a widely-used open source system monitor is used for application instrumentation, which collects measured data into a database of the central monitoring server. A Matlab program directly queries these data logs from the central database for post-processing, reaction planning, and execution

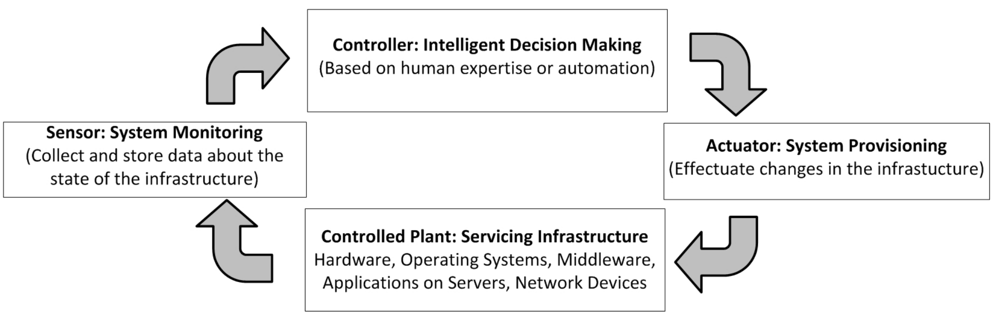

Figure 1: IT infrastructure management as a control loop

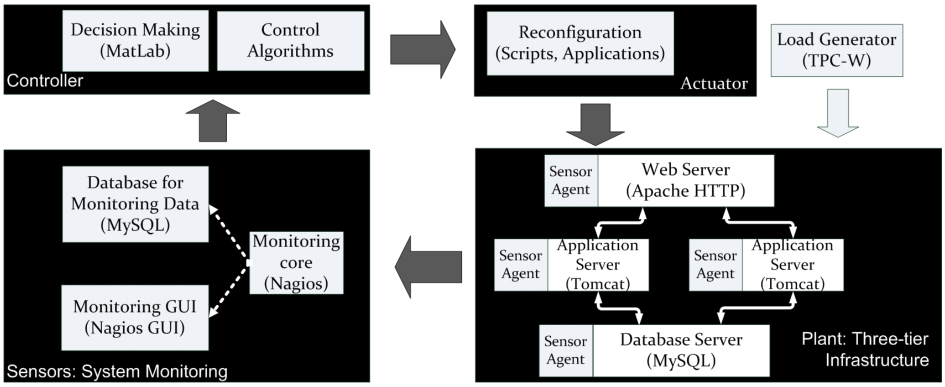

Figure 2: Three-tier infrastructure with monitoring and data processing

triggering. The three-tiered pilot TPC-W web benchmark infrastructure is composed of a web server, a runtime reconfigurable, load-balanced dual application server cluster, and a database server. A number of emulated browsers serve as workload generators according to the TPC-W specifications.

III. Initial results on large-scale IT infrastructure control

Some core problems of large-scale IT infrastructure control were addressed at first: sensor selection for data processing [3], and large-scale monitoring- and experiment-based simulation [4], transaction tracking in datacenters for workload characterization [5], and applying control-schemes to lossless datacenter networks to achieve latency and robustness goals [6]. [3] compares different approaches to the problem of selecting a small subset, out of the huge set of sensors offered by monitoring tools, still faithfully characterizing the system for control purposes. We concluded that linear methods methods are more accurate in 'normal' states of operation, while entropy-based approximations yield better results in critical 'degrading' states (which is critical for predicting and possibly avoiding overload). [4] investigates the analysis of datacenter infrastructures by means of analytical and simulation methods, and we argue that rigorous datacenter design and control benefit from full-scale simulations already proven useful in high-performance computing [7], but require advanced monitoring and modelling.

IV. Conclusions and future work

The presented experimental infrastructure for large-scale control development is functional. We will continue with detailed evaluation, system identification, and – on this basis – model-based control. Validation of the scalability of the empirical results gained in the framework to large-scale datacenters will be done by simulation-based model analysis [4, 7] parametrized by experimental data.

References

[1] M. Parashar and S. Hariri, *Autonomic computing: concepts, infrastructure, and applications*, CRC Press, 2006.

[2] J. Hellerstein, Y. Diao, S. Parekh, and D. Tilbury, *Feedback control of computing systems*, IEEE, 2004.

[3] G. Paljak, I. Kocsis, Z. Egel, D. Toth, and A. Pataricza, "Sensor Selection for IT Infrastructure Monitoring," in *Third International ICST Conference on Autonomic Computing and Communication Systems*, 2009.

[4] M. Gusat, C. DeCusatis, C. Minkenberg, L. McKenna, K. Bhardwaj, G. Paljak, A. Pataricza, and I. Kocsis, "Benchmarking the Ethernet-Federated Datacenter," in *Data Center Converged, Virtual Ethernet Switching Workshop*, 2009.

[5] G. J. Paljak, "Transaction tracking in large scale datacenters," Tech. Rep. RZ3743, IBM Research Zurich, 2009.

[6] M. Gusat, C. Minkenberg, and G. Paljak, "Flow and congestion control for datacenter networks," Tech. Rep. RZ3742, IBM Research Zurich, 2009.

[7] C. Minkenberg and G. R. Herrera, "Trace-driven co-simulation of high-performance computing systems using omnet++," in *2nd International Workshop on OMNeT++ (hosted by SIMUTools 2009)*. ICST, 2009.