

Public Review: Towards autonomic hosting of multi-tier Internet applications

Swaminathan Sivasubramanian, Guillaume Pierre and Maarten van Steen

Public Reviewer: Jeff Kephart

IBM Watson

Sivasubramanian, Pierre and van Steen discuss an approach to dynamically re-configuring a multi-tier system so as to best meet a specified service level agreement (SLA). The authors step beyond previous work in that they treat the multi-tier system holistically, and account for the common practice of caching at various tiers. The authors main point seems to me to be an almost self-evident truth: in order to drive the behavior of a multi-tier system in accordance with a given SLA, one must use optimization in conjunction with a reasonable model of how the service-level metrics depend on system control parameters and workload. The authors illustrate their ideas by means of a simple, easy-to-understand scenario. They parameterize the system configuration as the number of servers in each of an established set of four tiers with fixed interrelationships. Within the tiers are server-side, client-side, and database caches. Likewise, they introduce an appropriately simple service level metric: the response-time for a single service class. The authors present a parameterized model that expresses how this service-level metric depends on the system configuration and the workload. It is derived by considering how the end-to-end response time results from complex interactions among the various tiers, including execution times and queuing delays as well as the probability of cache hits and misses. The authors give a plausible account of how the various model parameters can be estimated. For example, Little's Law is employed to determine how adding or removing a server will affect queuing delays, and a virtual cache list is maintained to estimate how the cache hit ratio might be affected by more or fewer cache servers. When an SLA is violated, a controller consults the model to determine the tier to which a new server should be added so as to attain the greatest possible improvement in the SLA. The addition is made, and the process continues until the SLA is satisfied. Since this method can lead to a system that is overprovisioned in times of light workload, the controller must be vigilant for conditions in which the SLA is oversatisfied by some specified amount. The same procedure is then used to remove servers from tiers judiciously. The next step for this work is to mature from a plausible scenario to a plausible demonstration of the methods efficacy in more

realistically complex environments. I suggest three lines of investigation, any of which would be worthy of a paper at ICAC 2007 or a comparable venue:

1. **More complex SLAs.** I am curious to see how the authors will cope with SLAs that encompass multiple service classes and (later) availability criteria. Can they trust an established workload manager to manipulate the finer-grained low-level resources such as CPU and memory shares? Or are present-day workload managers focused too myopically on a single tier? In the latter case, can the authors use their approach to manage workloads holistically across tiers? This could entail including many more low-level parameters in the model and the optimization.
2. **Switching delays.** The authors should address the delay inherent in switching a server from one role to another. When a database cache server is moved to the web tier, several minutes may be required to quiesce it and install the new middleware and applications upon it. If the model and optimizer fail to account for these delays, the controller could thrash, and servers could spend half of their time being installed and de-installed rather than processing workloads.
3. **Multiple web applications.** I would like the authors to extend their methodology to a large-scale data center that hosts multiple applications. In this case, having applications voluntarily set response-time thresholds below which they are willing to give servers back is inadequate, as it unnecessarily throttles resource usage when resources are plentiful, and provides no conflict resolution when they are scarce. I suggest switching to SLAs that are based on utility functions, an approach that has been advocated and explored in papers appearing in all of the ICAC conferences to date.

While there is much work left to do, I believe this is a promising paradigm for managing autonomic computing systems (multi-tier or otherwise) to service-level objectives.