Explicit Performance Assurance for Virtualized Applications: Interference-Sensitive Specification, Placement, and Resource Allocation

NSFCloud Workshop on Experimental Support for Cloud Computing Position Paper

Alex Blate (blate@cs.unc.edu)
Department of Computer Science
University of North Carolina at Chapel Hill
October 2014

Abstract:

Applications deployed on traditional bare-metal infrastructures enjoy well-defined structural and resource semantics – they use dedicated hardware, their physical structures are readily-apparent, and they have limited, stable interactions with other applications via shared resources. When applications are migrated to or implemented on virtualized infrastructures – i.e., as virtual machines (VMs) – much of this structural information is lost or becomes hopelessly obfuscated. Moreover, the VMs comprising the application are exposed to interference via resources shared with other VMs and interactions with the hypervisor. As VMs are added, removed, and migrated within an infrastructure, the specific impacts upon application components change. The net effect is a new equivalence class of performance non-determinism in Virtualized Applications (long-lived collections of guests providing a set of end-user functionality) due to interference via shared resources. This can make it difficult to deliver consistent performance characteristics for such applications over time, hampering the adoption of virtualization for performance-sensitive or real-time-interactive applications and increasing the management costs of a larger set of applications. Our research is targeted at providing consistent performance to each individual Virtualized Application (VA); more specifically, we aim to minimize inter- and intra-application interference and to ensure that each VA, considered as a first-class entity, is placed and receives resources in accordance with the VA's structural and semantic requirements. The goal is to provide consistency of performance on par with what would be expected in bare-metal deployments. We introduce rich semantics for specifying the structure, resource requirements, interference sensitivities and effects, and dependencies of whole applications; such specifications enable automated, interference-sensitive placement and resource assignment and the structurally- and semantically-informed analysis of application telemetry. As an application's resource requirements are observed over time, their specifications may be iteratively-refined, providing a closed feedback loop converging upon the requisite levels of performance assurance. Specifications are also reusable for subsequent instances of the same application in other competent infrastructures.

Research Objectives:

- Identify, characterize, and quantify sources of interference, including those inherently introduced by virtualization (e.g., opaque and implicit shared resources).
- Develop generalized semantics for specifying VAs' structure and resource requirements – including those pertaining to interference.
- Develop placement algorithms, informed by said requirements, such that inter- and intra-VA interference is mitigated.
- Empirically demonstrate that interference mitigation leads to greater and more predictable application performance and that such mitigation is effectively achieved by such placement.
- Leverage the same specifications to drive structurally- and semantically-informed monitoring and analysis and the iterative refinement of resource requirements, dependencies, and sources and sensitivities to interference.

Relevance to NSFCloud program:

Enabling the performance assurances described above requires basic research into the characterization of components' (e.g., a VM's) use of infrastructural and virtualized resources and, in particular, characterizing sensitivities to inter-component interference via shared resources. Characterizing and encoding such requirements and performing component placement and deployment such that they are met remain open problems both in the literature and in industry. Interference is a complex function of applications themselves, how each application is assigned resources (which we will refer to as placement) and the assignment of resources to other applications. We must determine potential points of interference, how to identify them, how to characterize them for a particular application, and how to express them succinctly and quantitatively as input into, e.g., placement algorithms.

Fundamentally, we wish to have access to a representative corpus of real-world applications targeted at real human end-users. Each application's end-user-centric performance should be sensitive to one or more sources of resource interference, e.g., CPU, cache, storage, network latency, etc. We must be able to monitor and measure low-level application behavior, vis-à-vis resources in an environment free from interference with other experiments or activities. Initially, this enables the precise observation and measurement necessary for classifying and quantifying resource requirements and interference sensitivities. Subsequently, we desire objective end-user feedback to judge the efficacy of our placement mechanisms in comparison to common present-day placement practices.

We see significant value exploring our infrastructural and application requirements in the context of the NSFCloud program and the present Workshop: our application model and research objectives address an underexplored but highly-relevant class of use cases in cloud computing; the existing experimental infrastructures we have evaluated lack the fine-grained and low-level control, long-lived application model, or isolation parameters we seek; we are excited to work with application developers and designers – both to identify candidate applications and to leverage our platform and performance analytics to improve their applications; and, because our research is immediately applicable to present-day IT challenges, we see potential for attracting or enhancing industry collaboration and/or funding of the NSFCloud program as a whole.

We are presently collaborating with another research team at UNC-Chapel Hill who are pursuing NSF-funded (including CRI) research into wearable computing devices, a la Google Glass. We expect to host some of the back-end components of their applications, many of which have real-time interactive requirements. In the longer term, we see potential for research into the allocation and sharing of new resource types, such as virtualized GPU resources. We are also exploring collaboration with RENCI (the Renaissance Computing Initiative) to host certain front-end components for web-enabled applications.