Shirako: Virtual Machine Hosting for Networked Clusters
(presented with demo)

Laura Grit, David Irwin, Aydan Yumerefendi, Jeff Chase
Department of Computer Science
Duke University
{grit,irwin,aydan,chase}@cs.duke.edu

After a decade of advances in virtual machine (VM) technology, robust and efficient VM systems are widely available and and are fast becoming ubiquitous. Among other benefits, these systems offer powerful mechanisms for managing shared server networks and clusters. The leading VM systems support live migration, checkpoint/restart, and fine-grained allocation of server resources as a measured and metered quantity. Each VM is bound to a performance-isolated "sliver" of host resources; slivers are sized along multiple dimensions (e.g., CPU cycles, memory, and network bandwidth), and can be resized on-the-fly to adapt to changing demands. These capabilities create a rich policy space for system management infrastructures. How should an intelligent infrastructure "turn the knobs" to map workload and resource requests onto a server network? A key challenge today is to develop the policies for efficient and practical resource management using virtualization technology—on-demand, adaptive, and reliable allocation of networked computing resources from a common pool.

Management policy is a difficult challenge for at least three reasons. First, resource management involves projections under uncertainty and optimization problems that are NP-hard in their general form, forcing us to adopt heuristics tailored for specific needs and settings. Second, these policy choices must balance the needs and interests of multiple independent stakeholders. For example, resource consumers request resources according to their projected needs, while resource providers should arbitrate requests to allocate resources to their highest and best use. In federated systems—in which independent providers contribute resources to a shared pool—brokering intermediaries may also play a role to supplant the need for pairwise peering arrangements. In general, the resource assignment emerges from the interactions of policies in each of these actors. Finally, the assignments are dynamic: they must adapt to changing workload and demands.

We are exploring this policy space using Shirako, a Java toolkit for *resource leasing* services based on the SHARP framework. Although Shirako's leasing model can generalize to many kinds of shared resources (e.g., network resources), we are currently focusing on using it as a basis for secure, adaptive, on-demand resource sharing in federated clusters. In this setting, the policies control VMs *provisioning* (sizing) and *placement* of VM images within the server network.

Because the policy space is so complex, there is no "one size fits all" solution. Shirako defines interfaces for pluggable policy modules for each actor. A key principle is that provider sites control the placement of VMs on their own clusters, but delegate limited provisioning power to brokers, which may run on behalf of third parties and may arbitrate lease requests for multiple providers. This decoupling offers a general and flexible basis for managing a shared pool with many resource contributors: each broker and provider site is free to select the policies governing the resources under its control. For example, brokers can assign resources across multiple sites in a coordinated way, while sites can control placement of the hosted VMs based on local considerations, e.g., thermal load balancing within a cluster.

When VMs are provisioned and resized dynamically, a cluster site may be forced to adjust its placement choices by migrating VMs. This can occur, for example, if VM slivers grow in response to load changes in a hosted service, or if requests for new VMs are granted. We have extended Shirako to enable site policies to control migration and checkpoint/restart for Xen VMs. If the broker is correct and honest, then some feasible placement will always exist to satisfy its provisioning choices. To preserve accountability, the broker annotates its directives with a verifiable proof that its sequence of VM sizing choices always has a feasible assignment at each site. In general, computing a plan to minimize the migrations is NP-hard. We are exploring a home-based migration scheme that restores a feasible placement in a bounded number of steps, at the price of reserving memory at each VM's home node.

The Demo. Shirako has a Web-based portal interface to observe and operate on actors running at a given node. The demo uses the Web interface to create new actors, select from a set of policies for them, instantiate hosted application services from a set (batch job scheduler, Globus grid), subject those services to a predefined load, and observe adaptive VM hosting driven by the interacting policies.