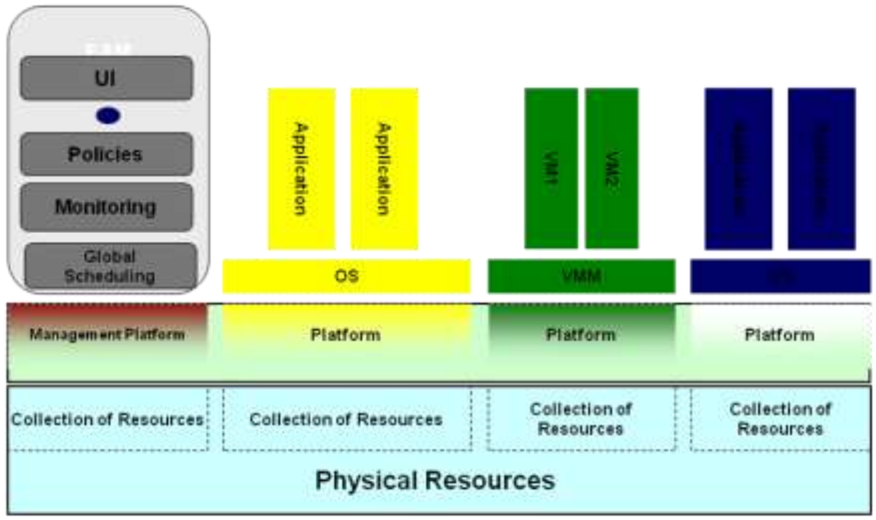
Scale-Right Provisioning for Next Generation Data-centers

by Salim Hariri

Scale-Right Provisioning (SRP) is intended to streamline deployment, allocation, management of compute, memory and I/O resources and scaling to efficiently react to runtime changes (e.g., failure, and workload). Adopting this will mitigate many of data-centers' inefficiencies such as: (*i*) Provisioning for peak loads and sometimes for multiple peak loads resulting in low average resources utilization (< 25%); (*ii*) Lack of autonomic features such as the inability to self-optimize when runtime workloads vary, inability to self-configure when resources are added/removed and inability to self-heal when failures happen; and (*iii*) Manageability, which has remained ad-hoc with considerable overhead on Total Cost of Ownership (TCO).



The premise of this project is to degenerate the physical resources (e.g., servers, network switches and SAN or NAS devices) in an enclosure (or server farm or rack or …) into *pools of virtual resources*. Then, based on the workload's resources requirements and its executions constraints such as Service Level Agreement (SLA), security and availability, a set of compute, memory, network and storage virtual resources are selected and launched as a dynamic virtual platform (Virtual Machine) to run the workload. Also, we provide the ability to automatically scale up/down resources allocated to a dynamic platform as runtime demands change. For example, when a running workload on a dynamic platform requires more compute power, one or more virtual logical threads from the resource pool is added in to this VM. Similar arrangements can be made for other scenarios of runtime changes such as power, SLA or availability. The specific features we plan to support include, but not limited to: (*i*) Ability to manage an enclosure of compute & I/O nodes independent of system software, and relying on high-level policies; (*ii*) Ability to create dynamic platforms from pools of compute/memory and I/O resources with capacities that match workloads' requirements; (*iii*) Ability to dynamically scale resources allocated to a dynamic platform up/down based on runtime changes (e.g., workload & failure); (*iv*) Ability to enforce various power budgeting schemes within the enclosure, as well as thermal-gradient characteristics; (*v*) Ability to mitigate failures through the integration of a Fault Prediction Agents to predict failures and gracefully migrate workloads from one dynamic platform to another; and (*vii*) Ability to perform fabric topology configuration to optimize resource allocation to dynamic platforms.

The vision of the capabilities provided by EVMM and EAM is to allow future backplanes to mimic current server motherboards. Specifically, what is on current servers' motherboards is chips (processors, chipsets, …) and a BIOS that enables them to expose themselves as a platform to run system software (OS or VMM). Similarly, EVMM enable of compute and I/O resources in various physical platforms to be aggregated and exposed as dynamic platform on which system software will run.

The Proof of Concept (PoC) for this project will include mechanisms to create: (*i*) pools of virtual resources; and (*ii*) a comprehensive autonomic management infrastructure. The first could be established through Virtual Machine Monitors (VMM) in each server or create an enclosure-wide VMM (EVMM). To create the EVMM, it is necessary to enhance existing VMMs with capabilities allowing them to communicate with each other and collaborate to project one aggregate VMM. Specifically: (*i*) Enable arbitrary grouping of resources to create dynamic platforms, and help create multiple of such platforms within an enclosure; (*ii*) Presents abstraction of the platforms to enable decoupling of hardware and software resources guaranteeing security and isolation; (*iii*) Help to dynamically allocate/de-allocate/reallocate resources within an enclosure and as signed to a platform; (*iv*) Helps manage the enclosure independently of system software; and (*v*) Helps redirect events to appropriate blade where a platform resides or to the management platform.

The autonomic management infrastructure includes multiple components: (*i*) Resource-Autonomic Manager (RAM) in each server with monitoring and limited decision capabilities based on locally collected data; (*ii*) Platform Autonomic Manager (PAM) with ability to make decisions at the platform-level; and (*iii*) Enclosure Autonomic Manager (EAM) that will have visibility to the whole enclosure. Specifically, the EAM is tasked with capabilities such as: (*i*) Self discovery and configuration of servers, inventory and cataloging capabilities at the enclosure level; (*ii*) Provisioning platform to boot OS images and applications based on system-level policies; (*iii*) Selecting virtual resources and launching dynamic platforms to run workloads; (*v*) Transparently migrating or scaling dynamic platform resources up/down based on runtime changes; (*vi*) Enforcing high-level policies such as those related to power and SLA; and (*vii*) projecting all information on a user interface to allow user to view both physical and virtual configurations in the enclosure.