vManage: Loosely Coupled Platform and Virtualization Management in Data Centers

Partial Review

Reviewed by

CIJO GEORGE
M.Sc (Engg.), SERC
S.R No. : 5510-210-101-07980
Email: cijo@rishi.serc.iisc.ernet.in
Important points

1. Introduction

- vManage: Loosely couples Platform Management and Virtualization management.
- Implements Virtualization aware Platform Management & Platform aware Virtualization Management.
- Uses already existing solutions for Platform & Virtualization Management.

2. vManage Solution

2.1 System Model

- All data collected by sensors are stored in cluster-level repositories.
- Registry & proxy mechanism: discovers, registers each sensor/actuator using publish-subscribe mechanism.
- Coordinator: implements control policies after getting information from registry & proxy.
- Stabilizer: ensures system stability.

2.2 Platform-aware Virtualization Management

- For Migration
  - Finds list of nodes satisfying resource requirements of the VM.
  - Gets current power consumption for these nodes from the repositories.
  - Power budget filtering is applied on the nodes based on this.
  - From the remaining nodes, a suitable node is selected for migration.

2.3 Virtualization-aware Platform Management

- SLA violation: increases frequency of CPU. If violation persists beyond a threshold, VM migration is triggered.
- SLA restore: decreases frequency of CPU

2.4 Stability for Coordinated VM Placement

- Stability of decision – Average Probability with host can continue to provide sufficient resources to the VMs for a given time into the future.
- Stability of system – Product of Stability of individual hosts.

3. Prototype Implementation

- Registry & proxy service uses CIM (Common Information Model)
- Each sensor/actuator registers with CIM Object Manager in their domain and are open to outside world through CIM classes.
- Repositories are implemented as LDAP directories.
- Coordinator is implemented in the application level as multi-threaded processes.
What I did not understand

- Some of the mathematics used in the design of Stabilizer is not clear.
- Doesn't this system require the modification of Xen to facilitate a new privileged driver domain Dom-M? Why can't the platform manager be implemented in Dom-0 itself?
- Is it not correct to say that the power consumed by a particular VM will be almost the same, which ever PM it resides in? If so, then how is migrating the VM helpful when power utilization is high in the PM? Doesn't the VM use the same amount of power in the new PM also, thereby making no change in the total power utilization of the datacenter?

What I liked

- The concept of trying to make virtualization management and platform management aware of each other is innovative.
- The simple system design, that can utilization already existing virtualization & platform management techniques without much changes, makes this model interesting.

What is missing

- SLA violations can occur due to bottlenecks in Network & Memory too. These factors are not considered by the Platform Management coordinator. First step that system proposed in this paper does on getting an SLA violation notification is increasing the CPU frequency. This may not be required at all if the bottleneck is in Network or Memory.
- The possibility of allotting more resources to the VM in the same PM to handle SLA violations (in case of non-work-conserving system), thereby trying to avoid Migration, is not looked into.
- In the experimental setup, only CPU resource type is considered during probability calculation in Stabilizer, and the stabilizer is said to have the knowledge of the workload model. So, it is not clear how the stabilizer performs when more resources are considered and no priori knowledge of workload is there.
- Scalability of the system is not explained. It is not clear how this system will perform in large datacenters.