oVirt SLA: MoM as host level enforcement agent

Shanghai 2013

Doron Fediuck
Red Hat

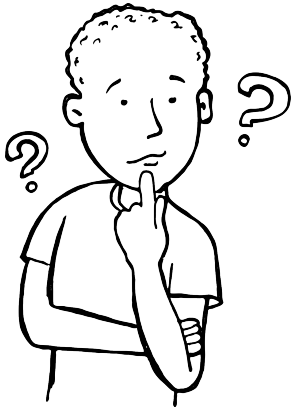
Overview

oVirt SLA fundamentals

Overview: SLA

oVirt

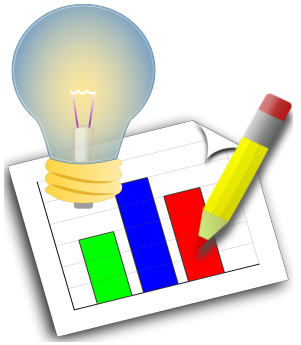
- **SLA**: Service Level Agreement
  - Ensures Quality of Service (QoS) based on parameters and a schema.



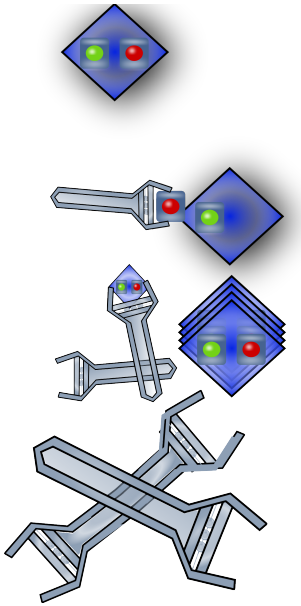
  - ISP
    - Schema would be Internet access.
    - Parameters: Up/Down bandwidth, ASA (Average Speed to Answer), etc.
- In Cloud computing this is becoming crucial, as we're providing IaaS

Overview: SLA

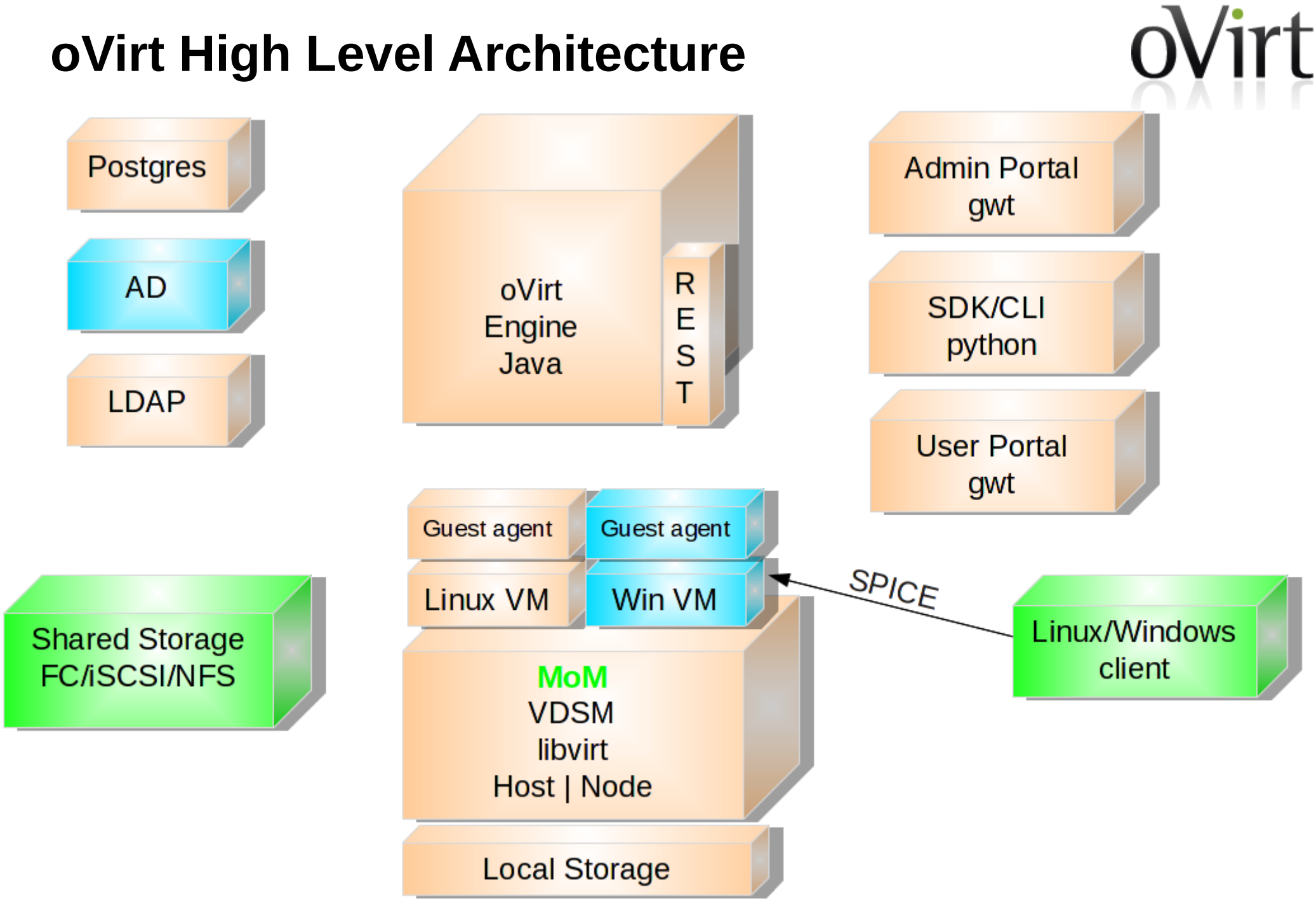
So what can we do for QoS?



- Gradually introduce SLA elements into oVirt
  - Add various features which will function as a toolbox
  - Improve MoM as an enforcement agent
  - Prepare the infrastructure for advanced SLA concepts
    - VirtIO Memory Balloon
    - KSM
    - Many more to come...



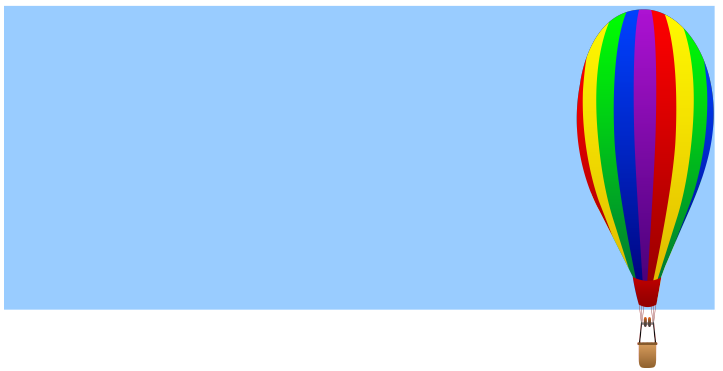
oVirt High Level Architecture



VirtIO Memory Balloon

- The balloon driver is a special process
  - Non-swappable and un-killable
  - May be inflated or deflated
- Inflate => take more RAM from the guest OS
- Deflate => return RAM to the guest OS

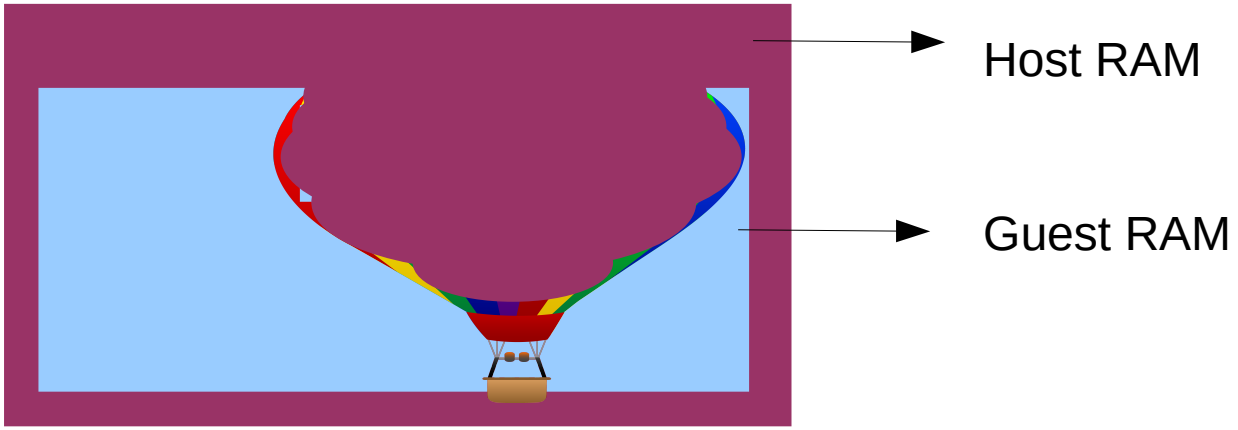
Free RAM for other processes



Free RAM for other processes

VirtIO Memory Balloon

- Memory pages in the balloon are unmapped
- Then, reclaimed by the host



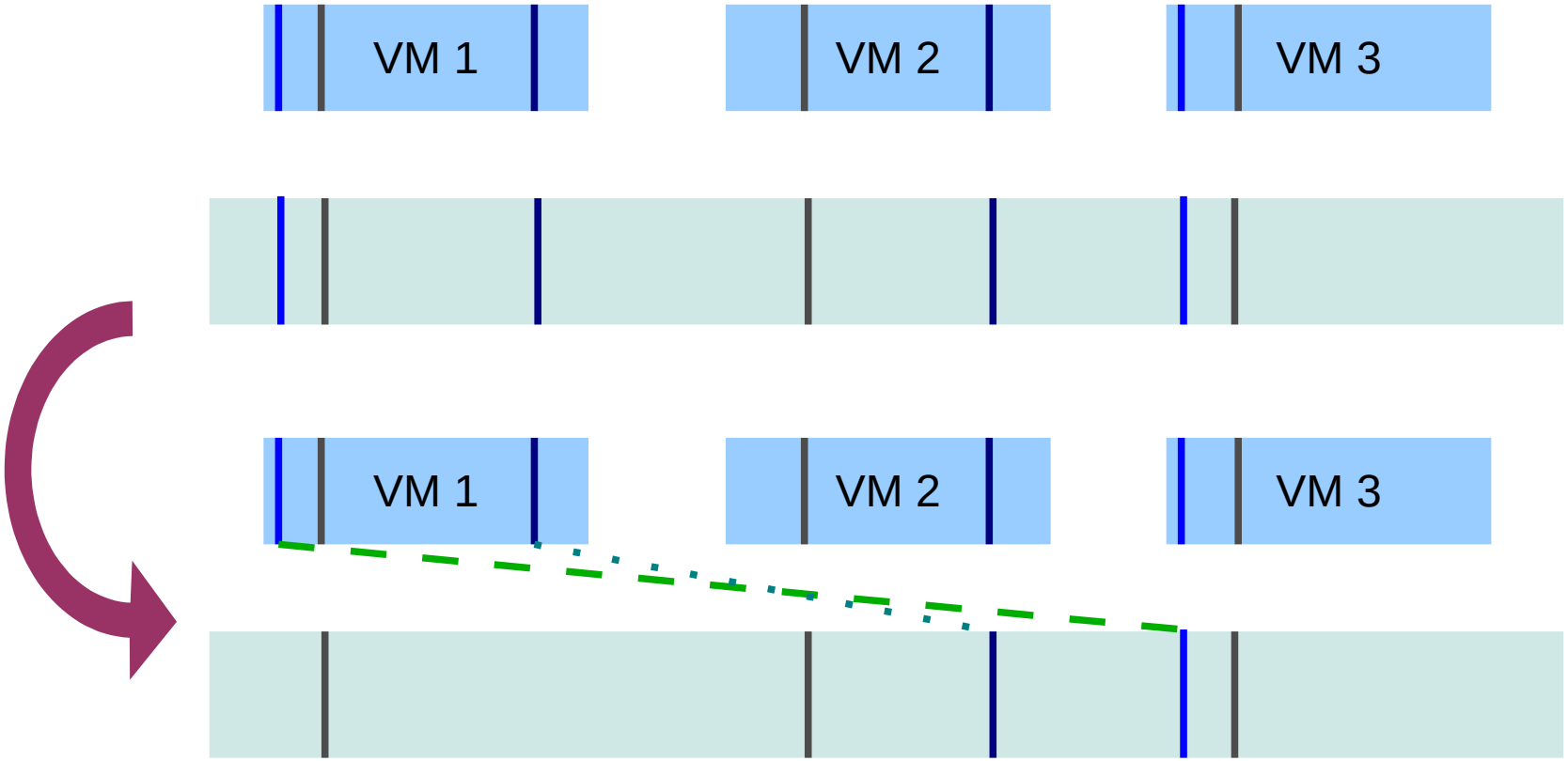
And now we can do memory over-commitment!

- 2 GB physical server runs 2x1GB VMs
- Using the balloon we can run 3x1GB VMs
  - Each VM's balloon will free 512MB back to the host

KSM

oVirt

- Kernel SamePage Merging

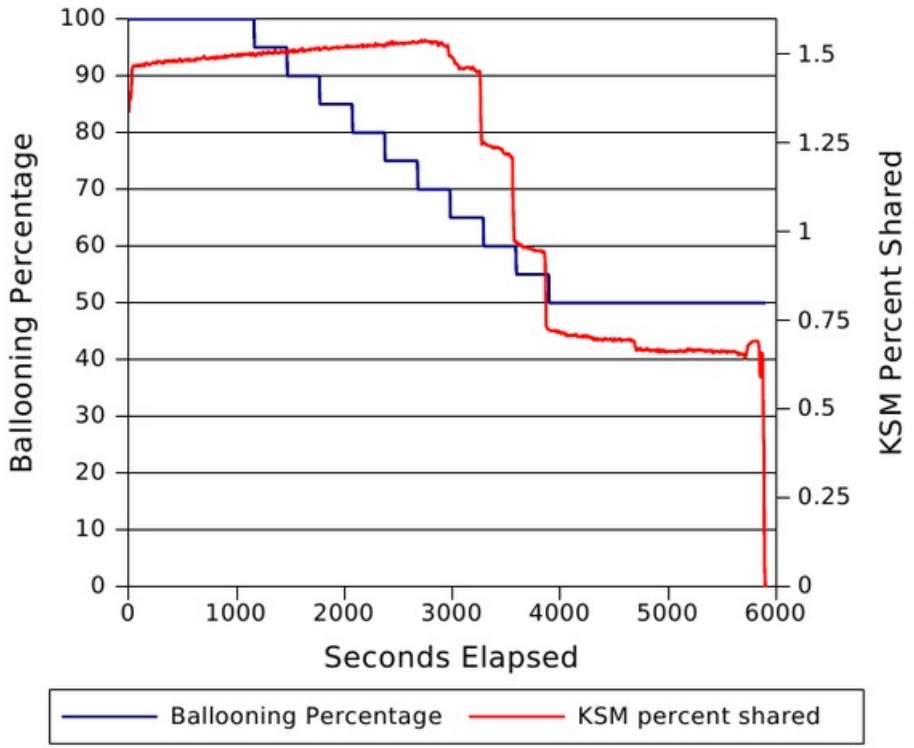


- 52 virtual instances of Windows XP with 1GB of memory, could run on a hypervisor that had only 16GB of RAM

oVirt

Host-level considerations

Host-level considerations

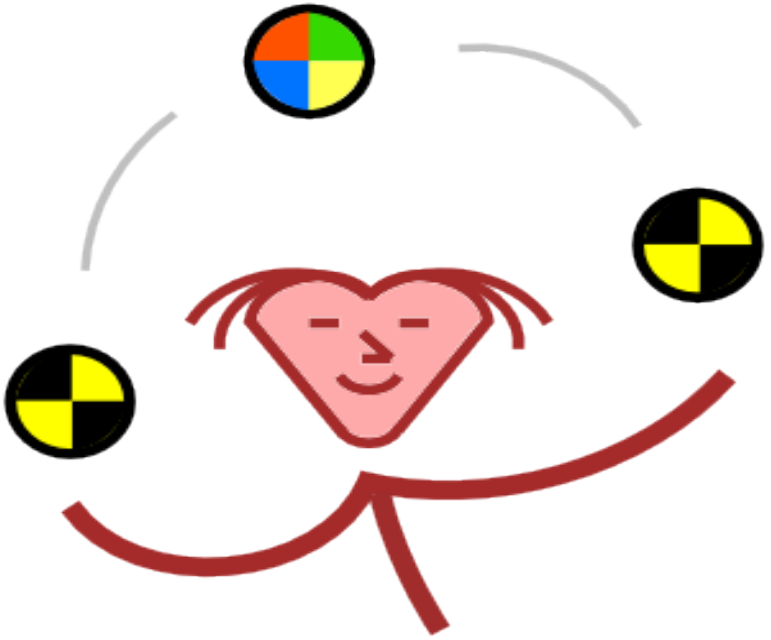


- Guest balloon drivers select pages to balloon without considering whether the host page might be shared.
- Ballooning a shared page is a mistake because it deprives the guest of resources without actually saving any memory

MoM to the rescue!

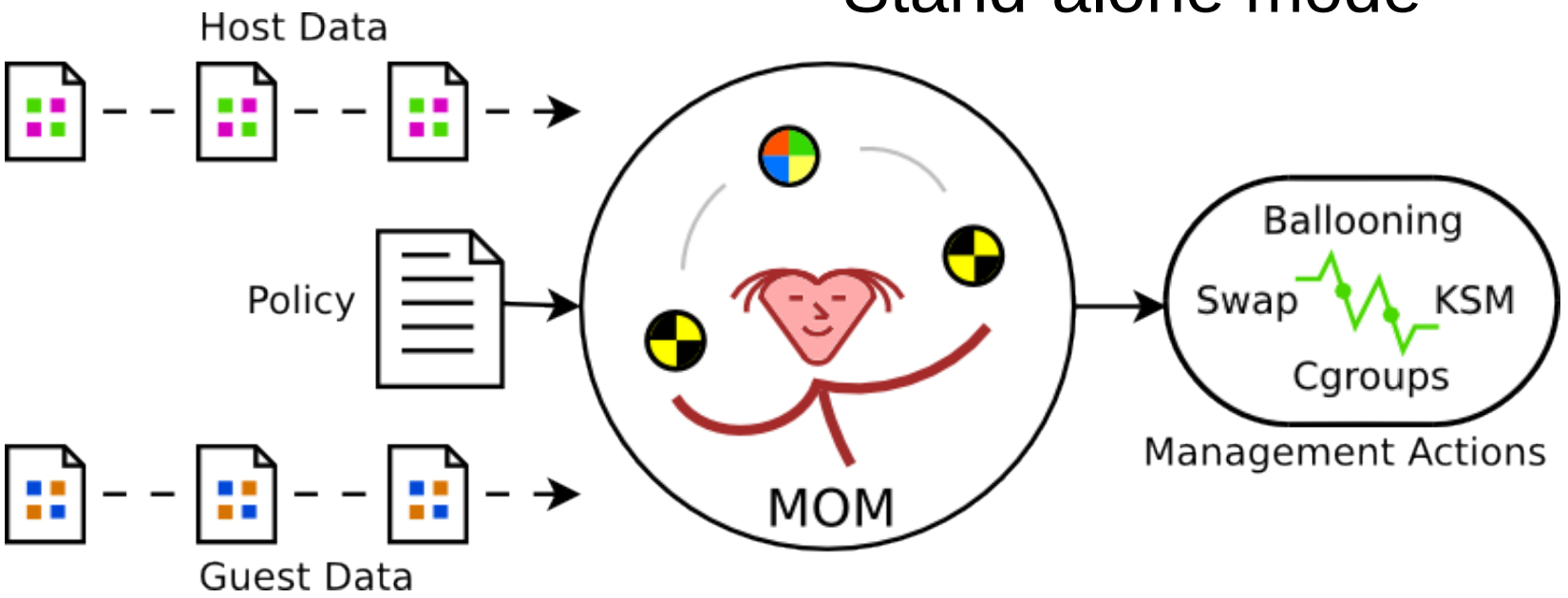
oVirt

- Written and maintained by Adam Litke (IBM)
- Joined oVirt as an incubation project
- Monitors and handles KSM and ballooning
- Trying to prevent interaction mistakes
  - Ballooning VS KSM

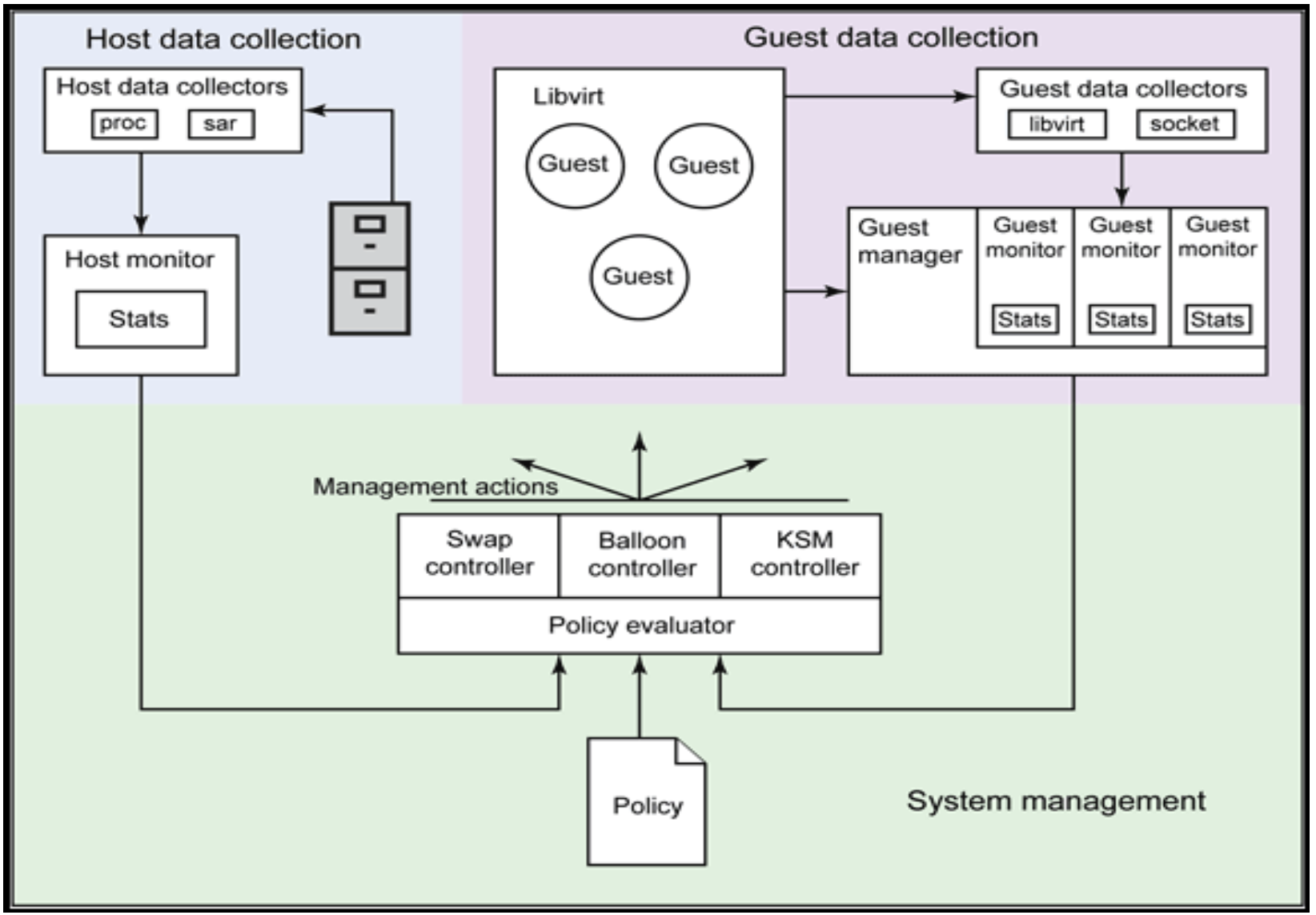


Introducing MoM

- Guest tracking
- Stats collection
- Fully extensible
- Dynamic policy engine
- Support for KSM and ballooning
- Stand-alone mode



MoM high-level architecture



MoM Policy Format

- Lightweight LISP-like policy language
- Access to stats and controls through simple variables
- No looping (except built-in guest iteration)

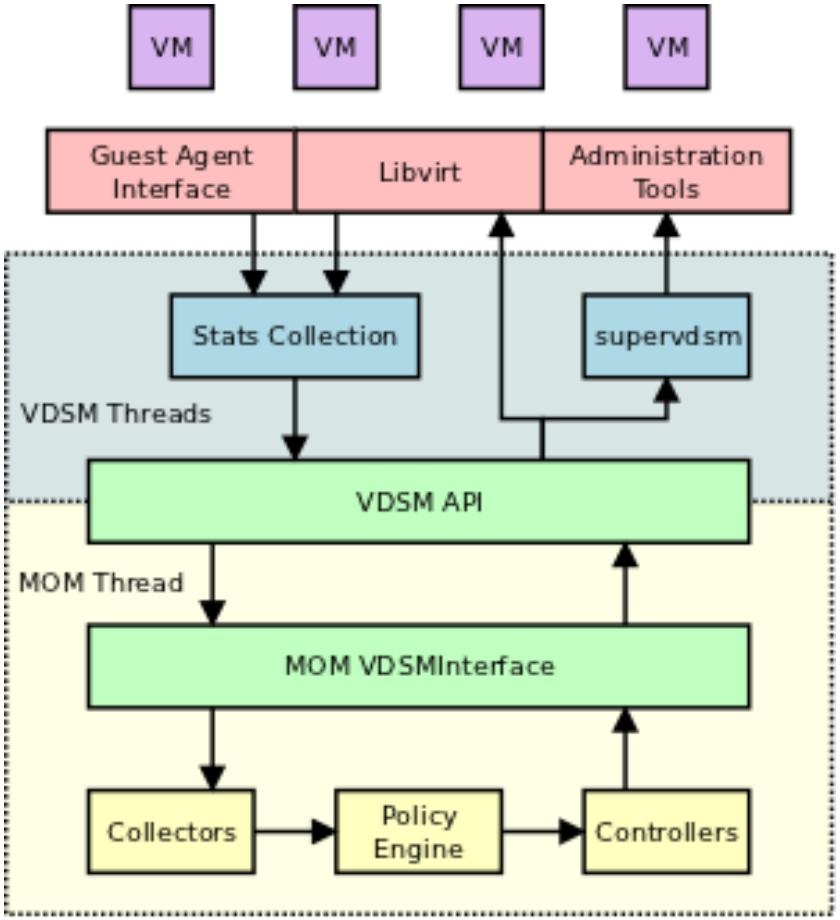
```
# The number of ms to sleep between ksmd scans for a 16GB system.  Systems with
# more memory will sleep less, while smaller systems will sleep more.
(defvar ksm_sleep_ms_baseline 10)

# A virtualization host tends to use most of its memory for running guests but
# a certain amount is reserved for the host OS, non virtualization-related work,
# and as a failsafe.  When free memory (including memory used for caches) drops
# below this parcentage of total memory, the host is deemed under pressure. and
# KSM will be started to try and free up some memory.
(defvar ksm_free_percent 0.20)

### Helper functions
(def change_npages (delta)
{
    (defvar newval (+ Host.ksm_pages_to_scan delta))
    (if (> newval ksm_npages_max) (set newval ksm_npages_max) 1)
    (if (< newval ksm_npages_min) (set newval ksm_npages_min) 0)
    (Host.Control "ksm_pages_to_scan" newval)
})
```

MoM-VDSM Integration: under the hood[1]

- MoM threads run within vdsm
- Stats collected via the vdsm API
- KSM / ballooning operations via vdsm API
- VDSM installs a default MoM policy



[1] http://wiki.ovirt.org/wiki/SLA-mom

oVirt

MoM: going forward

Current status

oVirt

MoM integration[1]

- MoM is the enforcement agent of oVirt
- VDSM integration done by Adam Litke and his colleagues (Mark Wu, Royce Lv)
  - Still gaps on engine side.

Starting oVirt 3.2

- Basic integration for KSM functionalities
- API support for memory balloon
- Packaging and maintaining (added to Bugzilla)
- Now adding capping (limitations) API support to VDSM
  - CPU & Memory (guaranteed, hard and soft limits)

[1] http://wiki.ovirt.org/wiki/SLA-mom

Work in Progress

oVirt

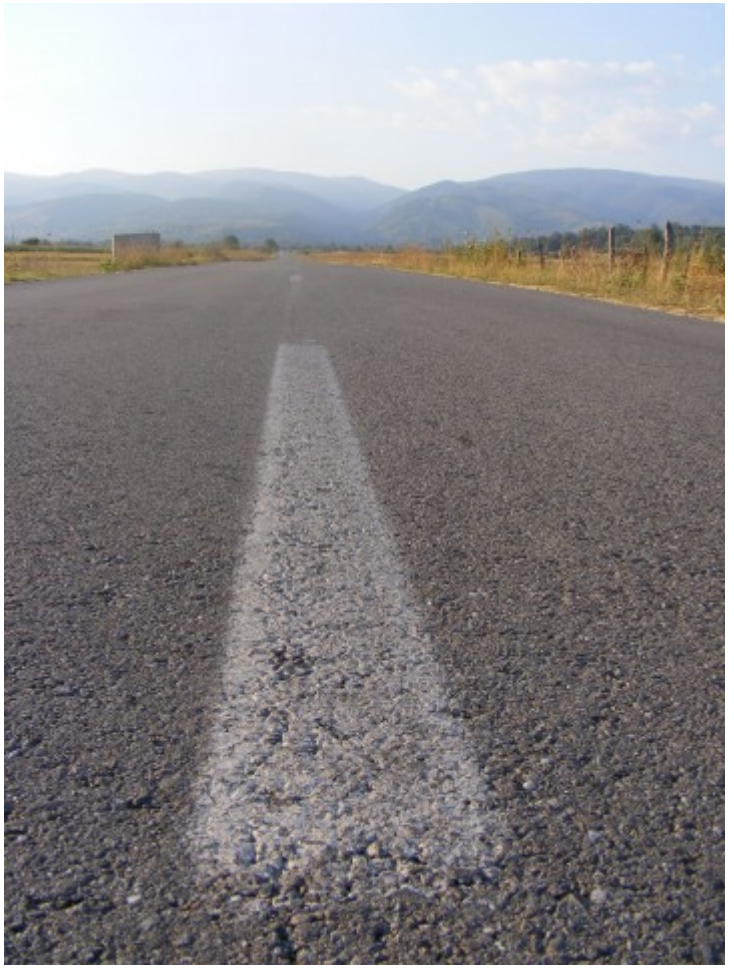
MoM integration[1]

- Fill-in gaps on engine side
- Now adding capping (limitations) API support to VDSM
  - CPU & Memory (guaranteed, hard and soft limits)
- **Considering various policies**
  - The biggest challenge: loads are changing
- Allow multiple policy parts
- More testings!

[1] http://wiki.ovirt.org/wiki/SLA-mom

oVirt SLA Road-map

- SLA features
  - Network QoS
  - HEAT integration (Application HA)
  - NUMA (numad, auto-numa)
- Extend MoM capabilities
  - Limitations for network & storage
  - Handle specific VMs
  - Additional policies
- MoM Continuous Integration



oVirt

and now is a good time for....

Questions?

oVirt

THANK YOU !

http://wiki.ovirt.org/wiki/Category:SLA
engine-devel@ovirt.org
vdsm-devel@lists.fedorahosted.org

#ovirt irc.oftc.net