KVM on Grid, Shaken, Not Stirred
Cloud batch scheduler integration of KVM

Raphael Sack

Systems programmer, Intel Corporation

Co-author: Igor Ljubuncic

IT@Intel

Legal Notices

This presentation is for informational purposes only. INTEL MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

* Other names and brands may be claimed as the property of others.

Agenda

- Introduction & background

- Overview

- Our solution

- Results & summary

- Future

Who am I?

- System Programmer, Engineering Computing

- Web and system related development

- Windows & Linux

- Car enthusiast

- Strong South African Accent :-)

PIC

(intel)

2014 Intel IT Vital Statistics

>6,300 IT employees
59 global IT sites

>98,000 Intel employees[1]
168 Intel sites in 65 Countries

64 Data Centers
(91 Data Centers in 2010)
80% of servers virtualized
(42% virtualized in 2010)

>147,000+ Devices
100% of laptops encrypted
100% of laptops with SSDs
>43,200 handheld devices
57 mobile applications developed

Source: Information provided by Intel IT as of Jan 2014
[1]Total employee count does not include wholly owned subsidiaries that Intel IT does not directly support

IT@Intel

(intel)

5

Overview of Intel chip design environment

- 40 sites globally

- 600,000 cores

- 5 PB memory

- 24 PB distributed storage

- Batch environment

- 40M+ regressions/month

(intel)

So what is KVM on a Grid?

KVM on a Grid

- A comprehensive solution

- Enables rapid transformation of the operating system landscape

- Spawn any task/OS at a click of a button (or command line)

- On demand provisioning

- KVM based

- Seamless addition of virtual machines to the environment

Why do we need it?!

Operating systems

- New major release every couple of years

- Time required to verify new image

- Time required to adapt

- Time required to switch over

- Different types of customer → different time tables

- Support cycle vs. how we do things (no time to change)

What can we do about this?

- Stay put, avoid changes

- Accept no new hardware and technologies

- Leave no option for your users/customers

- Maintain several images & machines simultaneously
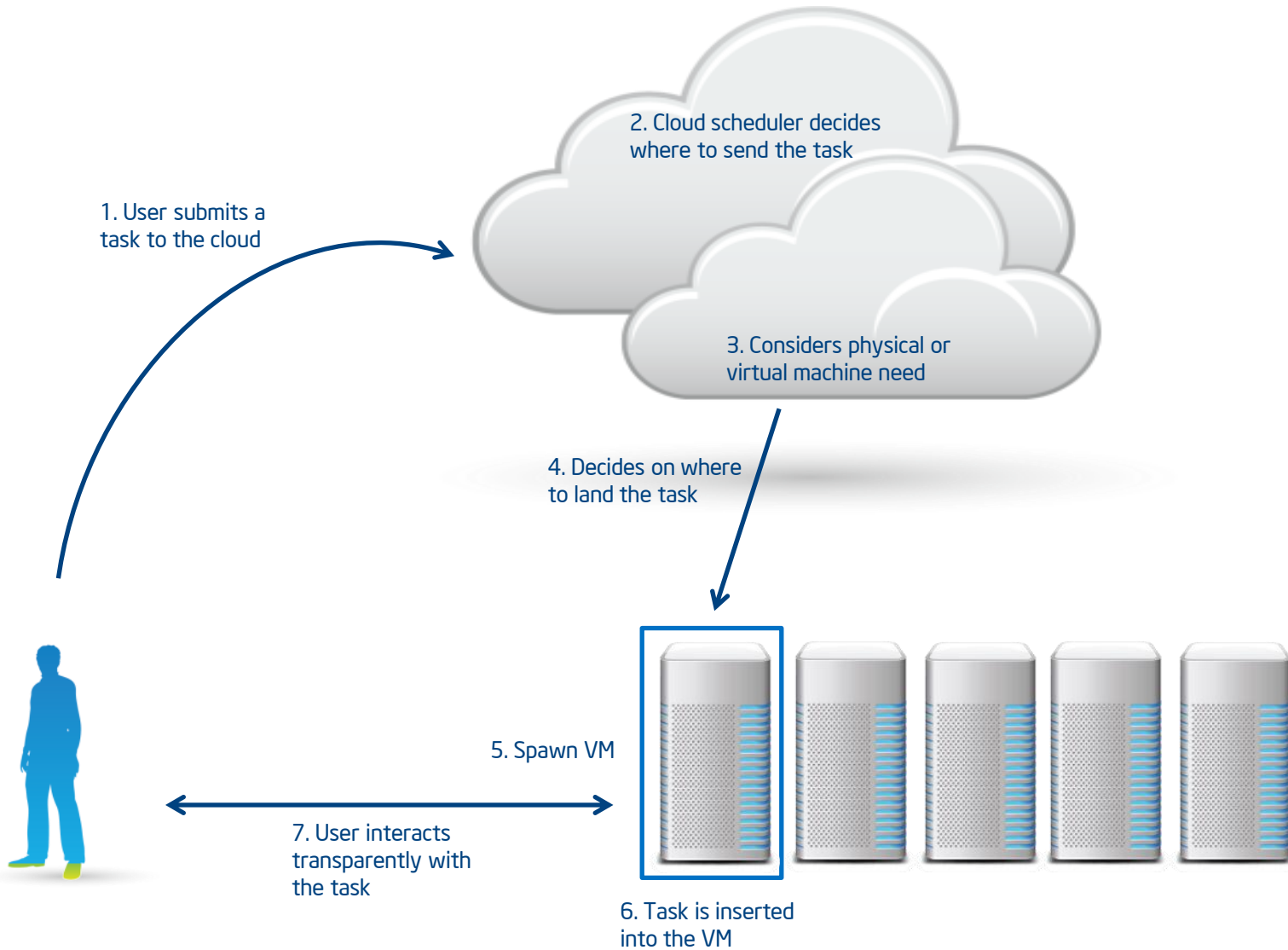
(intel)

Ideal state

- No specific OS release dependency

- Infinite on demand capacity

- Be able to change operating systems immediately

- One (operating system) size fits all

- Go visit the Nürburgring - Nordschleife!

(intel)

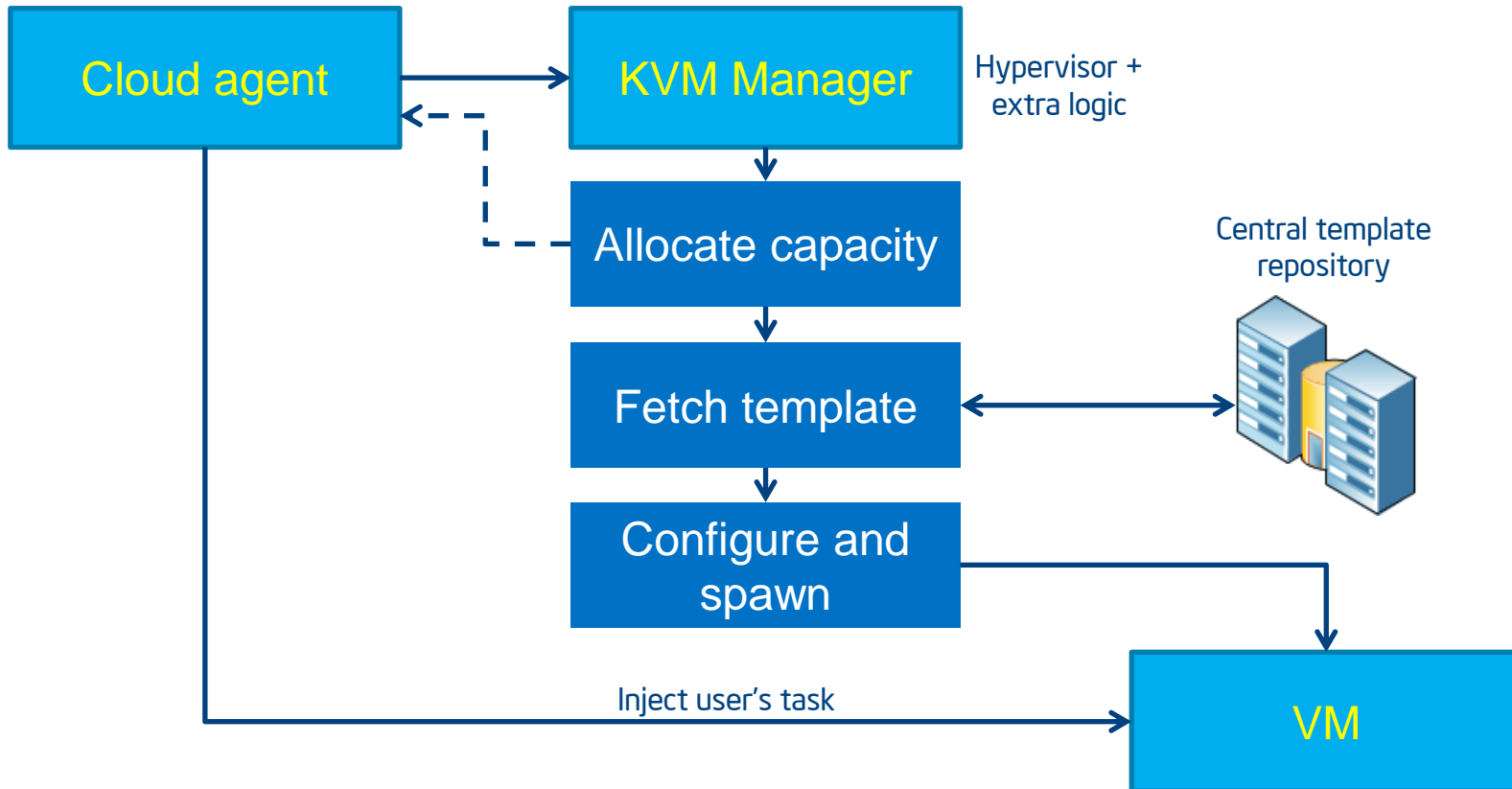# But seriously… we're dealing with customers

Our solution

- Extend the batch cloud system to support virtualization

- Make virtualization as transparent as possible

- Allow users to choose the required OS or choose for them

- Both for interactive and batch/non-interactive usage

- Limit use of older OS

- Run Windows on Linux :O

- "One Cloud"

Architecture

2. Cloud scheduler decides where to send the task

1. User submits a task to the cloud

3. Considers physical or virtual machine need

4. Decides on where to land the task

5. Spawn VM

7. User interacts transparently with the task

6. Task is inserted into the VM

(intel)

Architecture

Cloud agent → KVM Manager

Hypervisor + extra logic

KVM Manager → Allocate capacity

Allocate capacity ⇢ (dashed back to KVM Manager)

Allocate capacity → Fetch template

Fetch template ↔ Central template repository

Fetch template → Configure and spawn

Configure and spawn → VM

Cloud agent → VM : Inject user's task

Use cases

- Software validation

- Legacy OS & tools

- Throughput vs. performance

- Urgent short-term capacity needs

- Isolation and containment of users/tasks

- Network related restrictions and limitations

Results

- Major chip design business groups adoption

- Up to 3K virtual machines concurrently spawned (on demand)

- Mixed Windows and Linux based tasks

- 5M+ tasks in the past several months

- Dynamic capacity allocation

STOP!
Number time!

Results - Single task

	OS 1 Physical	OS 2 Physical	OS 1 VM	OS 2 VM
CPU*	1.00	0.80	1.04	0.81
IO	1.00	1.20	2.61	7.29
Memory Read	1.00	2.32	0.26	0.80
Memory Write	1.00	1.84	0.33	0.80

* Lower is better

Results - Multiple tasks

	OS 2 Physical	OS 1 VM	OS 2 VM
CPU*	1.00	1.28	1.27
IO	1.00	12.9	21.5
Memory Read	1.00	0.19	0.28
Memory Write	1.00	0.14	0.36

* Lower is better

Results - Hybrid

	OS 2 Physical	OS 2 VM
CPU*	1.00	1.01
IO	1.00	9.67
Memory Read	1.00	0.29
Memory Write	1.00	0.37

* Lower is better

Challenges

- Improving performance while minimizing cost

- Heterogeneous virtualization enclaves

- Increased complexity of implementation

- Disruptive changes to the environment

- Absolute necessity of transparency for users

- Support

Benefits

- Versatility

- Simpler and wider control of "old" images

- Significant cost avoidance

- Fully transparent

- Streamlined

- "Free"

Next steps

- Wider adoption by internal customers

- Performance optimizations and scale-out

- BI integration

- Live migration?

- Containers?

Summary

- Innovative use of existing KVM technologies

- Complex business case

- Challenging the status quo

- Increased flexibility in provisioning, deployment and OS migration

(intel)

Questions?

(intel)