

The nonkernel: A Kernel Designed for the Cloud

Muli Ben-Yehuda^{1,2}, Omer Peleg¹, Orna Agmon Ben-Yehuda¹,
Igor Smolyar¹, Dan Tsafir¹

¹Technion & ²Hypervisor Consulting Ltd.





Where is the money?



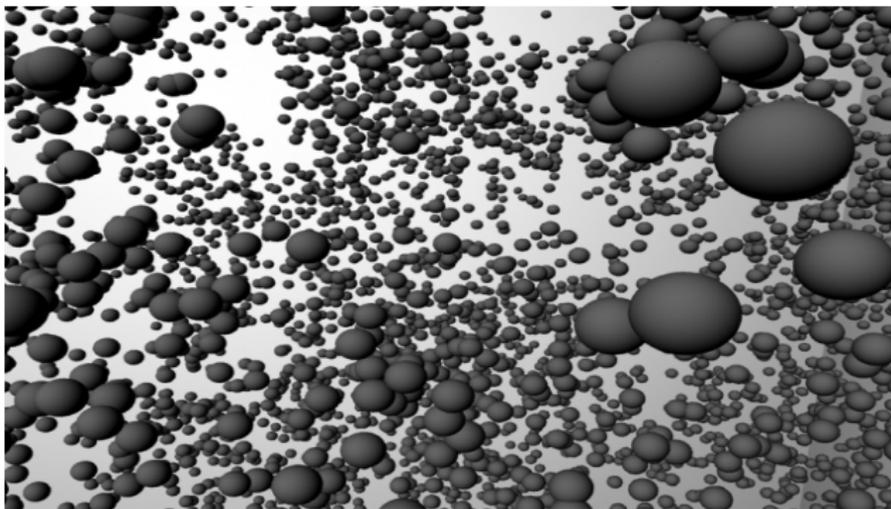
Where is the money?



In the clouds

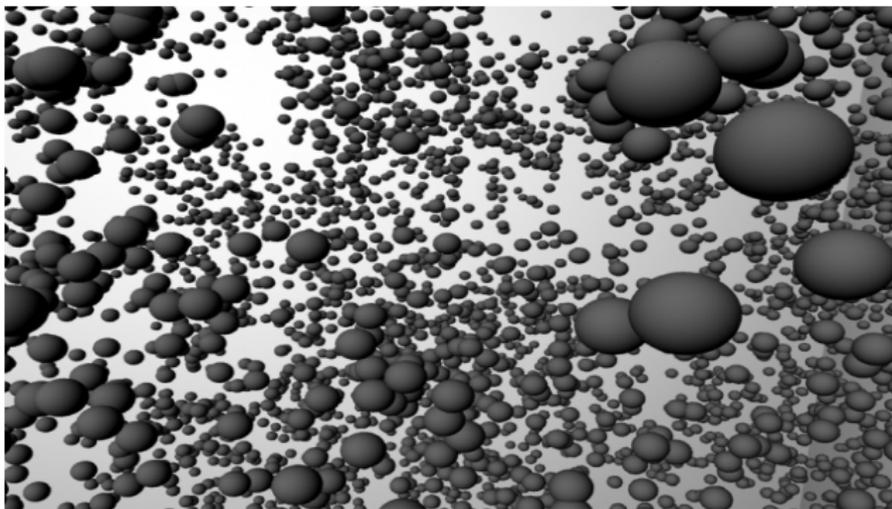


In the clouds



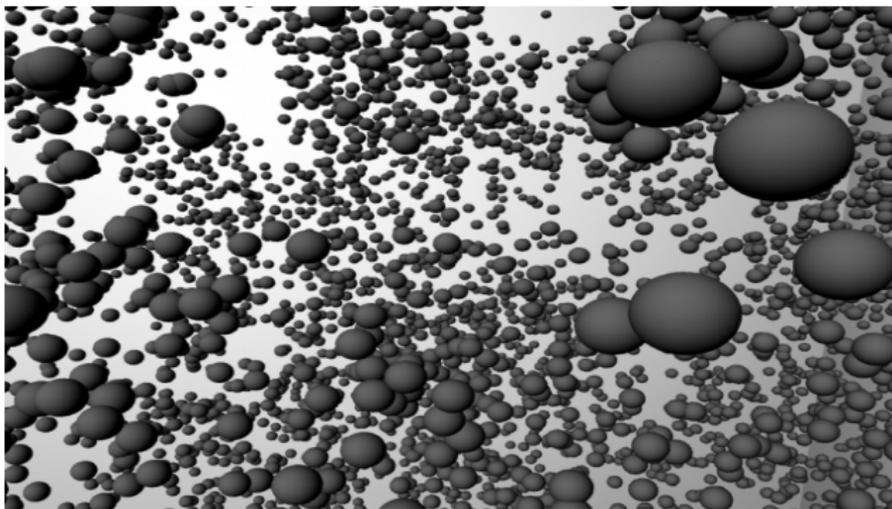
The Resource-as-a-service (RaaS) Cloud [[HotCloud'12](#)], [[CACM](#)]

- Renting fine-grained individual resources, not bundles (“server-equivalents”)
- For increasingly shorter time durations
- Market-driven resource pricing



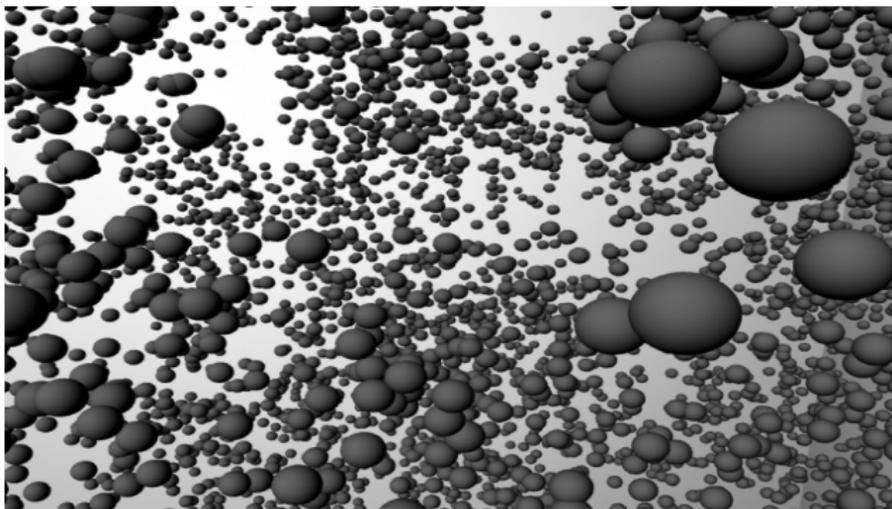
The Resource-as-a-service (RaaS) Cloud [HotCloud'12], [CACM]

- Renting fine-grained individual resources, not bundles (“server-equivalents”)
- For increasingly shorter time durations
- Market-driven resource pricing



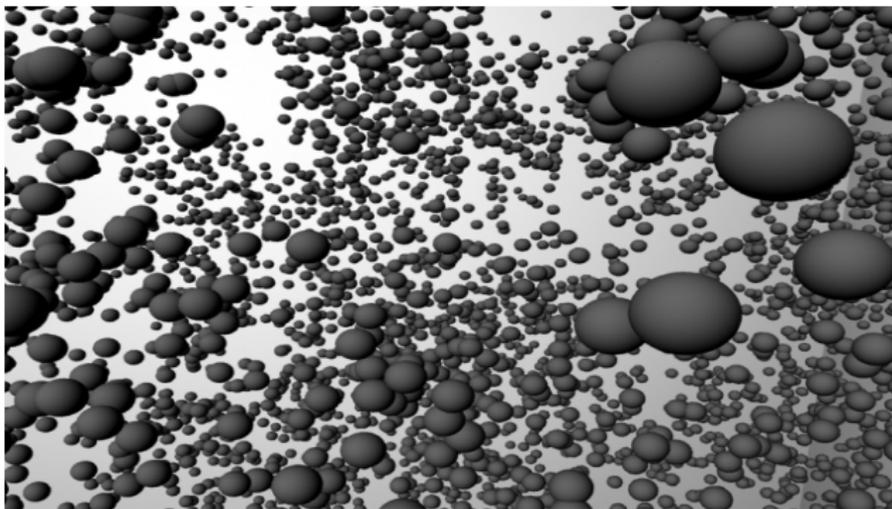
The Resource-as-a-service (RaaS) Cloud [HotCloud'12], [CACM]

- Renting fine-grained individual resources, not bundles (“server-equivalents”)
- For increasingly shorter time durations
- Market-driven resource pricing



The Resource-as-a-service (RaaS) Cloud [HotCloud'12], [CACM]

- Renting fine-grained individual resources, not bundles (“server-equivalents”)
- For increasingly shorter time durations
- Market-driven resource pricing



The Resource-as-a-service (RaaS) Cloud [[HotCloud'12](#)], [[CACM](#)]

- Renting fine-grained individual resources, not bundles (“server-equivalents”)
- For increasingly shorter time durations
- Market-driven resource pricing

Trend: Granularity of Duration of Rent



- 3 years on average: buying hardware
- Months: web hosting
- Hours: EC2 on-demand (pay-as-you-go)
- 5 minutes: CloudSigma, EC2 Spot Instances (pay-as-you-go)
- 3 minutes: GridSpot (as of July 2012)
- 1 minute: Profitbricks (as of July 2012), Google Compute Engine (as of May 2013)
- ...

Trend: Resource Granularity



- Amazon allows clients to dynamically change I/O resources.
- Since August 2012, Amazon also allows clients to set a desired rate on a per-block-instance basis.
- CloudSigma, GridSpot, and ProfitBricks offer clients to compose a flexible bundle—with prices depending on current cost of resources



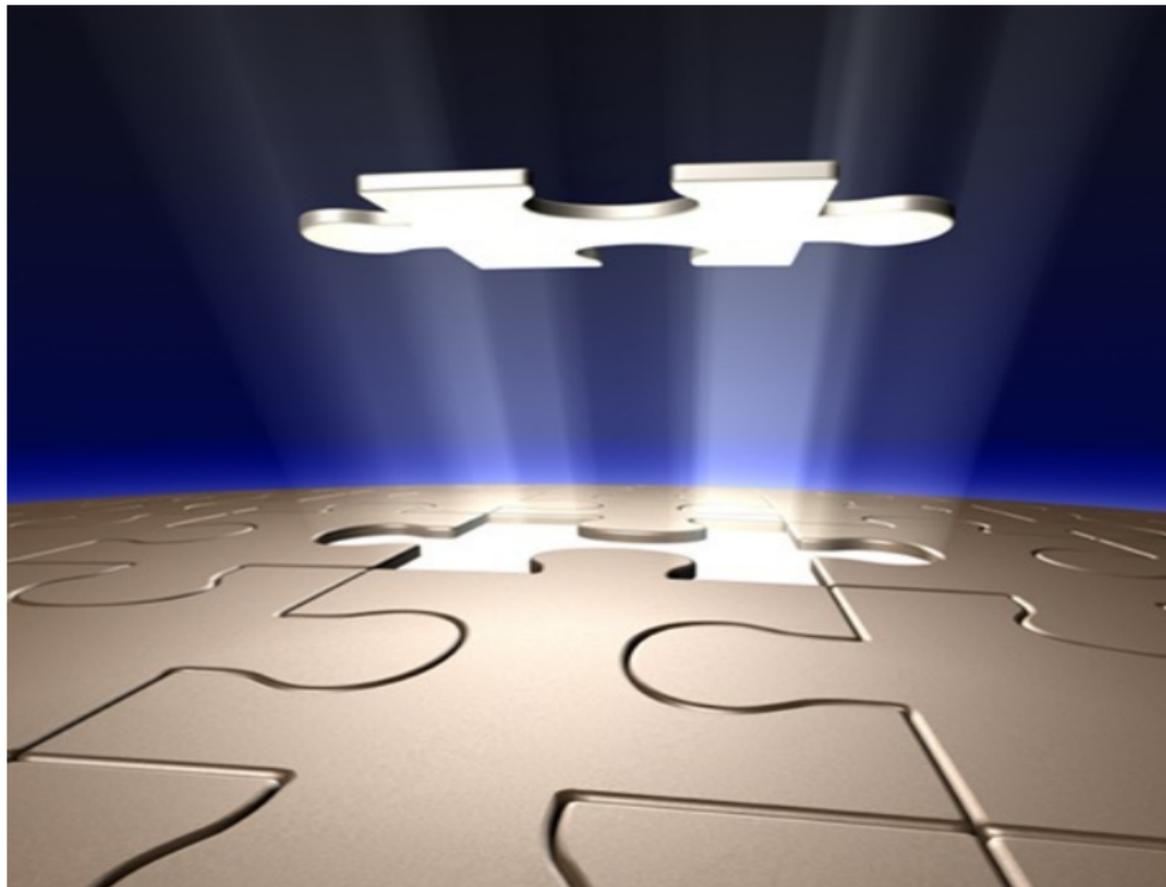
Unfortunately. . .



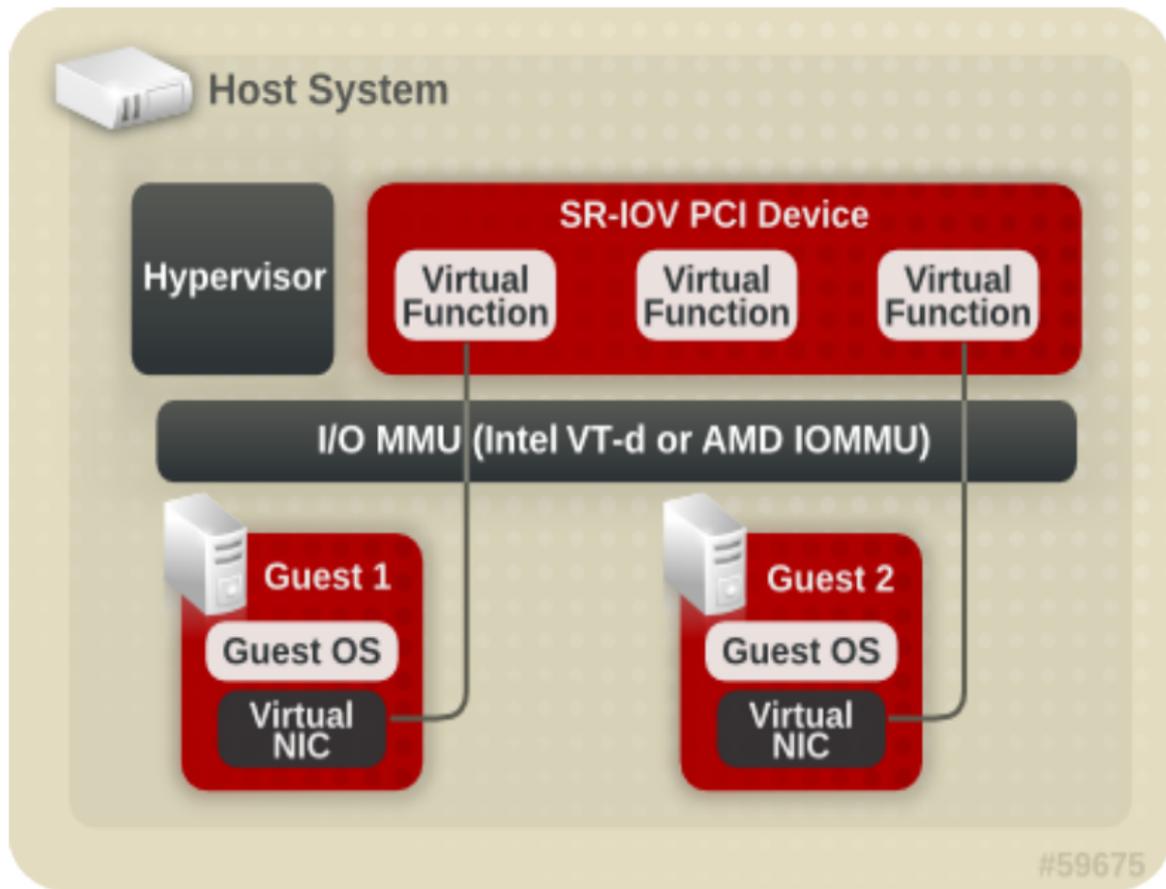
The cloud is different



- Resource ownership and control: the OS is no longer sole owner
- Economic model: provider vs. multiple clients
- Resource granularity: competition for resources \Rightarrow fine-grained resource allocation



The missing piece: architectural support for machine virtualization



Designing a cloud kernel: requirements



- Enable applications to bi-objectively optimize for cost and performance, not just performance
- Expose physical resources
- Get out of the way and let applications acquire, release and manage their own resources
- Isolates applications

Designing a cloud kernel: requirements



- Enable applications to bi-objectively optimize for cost and performance, not just performance
- Expose physical resources
- Get out of the way and let applications acquire, release and manage their own resources
- Isolates applications

Designing a cloud kernel: requirements



- Enable applications to bi-objectively optimize for cost and performance, not just performance
- Expose physical resources
- Get out of the way and let applications acquire, release and manage their own resources
- Isolates applications

Designing a cloud kernel: requirements



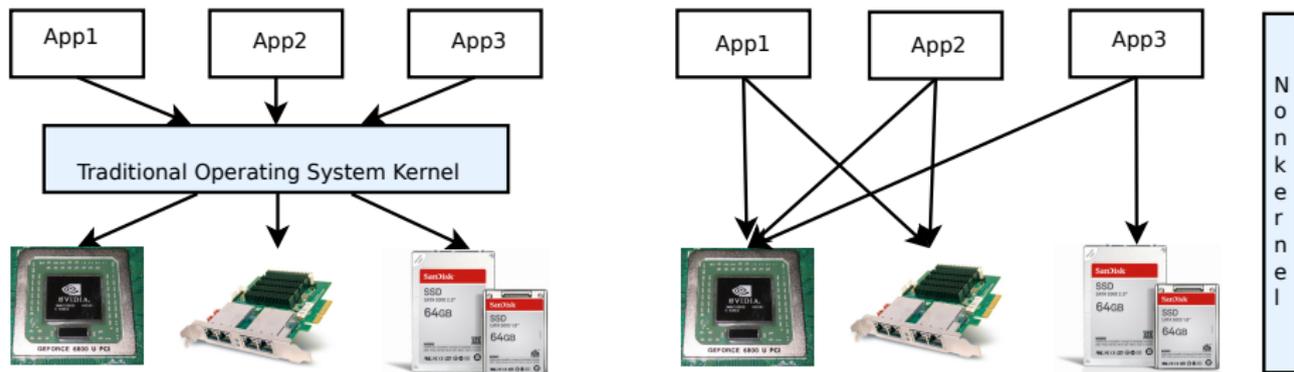
- Enable applications to bi-objectively optimize for cost and performance, not just performance
- Expose physical resources
- Get out of the way and let applications acquire, release and manage their own resources
- Isolates applications

Designing a cloud kernel: requirements



- Enable applications to bi-objectively optimize for cost and performance, not just performance
- Expose physical resources
- Get out of the way and let applications acquire, release and manage their own resources
- Isolates applications

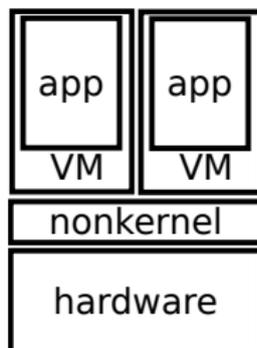
Welcome to the nonkernel!



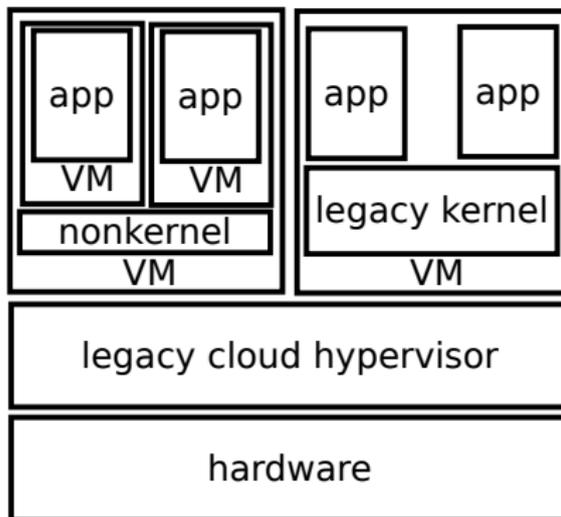
- The nonkernel is a hybrid kernel/hypervisor designed for the cloud
- Allows bi-objective optimization of both useful work and cost
- Exposes resources and their costs directly to applications
- Isolates applications from one another

The cloud software stack

The nonkernel
running
on bare-metal



The nonkernel on top of a legacy
hypervisor, alongside a legacy OS



Digging deeper into the nonkernel



- Hardware-assisted virtualization
- Minimal kernel: no device drivers or I/O stacks
- Applications know best how to manage their resources
- ⇒ All resource-related code is at application level



What the nonkernel does:

- Boots the machine
- Provides a clearinghouse for resources and arbitrates contended resources
- Isolates applications
- Efficient IPC without kernel involvement

Building a nonkernel



- An existing operating system / hypervisor? (e.g., with [Dune]?)
- From scratch?

Discussion: pros



- Performance
- Zero-overhead virtualization
- Reduced driver complexity
- A more secure system
- A more efficient system due to the economic model



- A clean break from the past
- No legacy hardware
- No legacy software

- Userspace I/O and virtual machine device assignment
- The Exokernel [[SOSP'95](#),[SOSP'97](#),[TOCS'02](#)]
- Library operating systems [[VEE'07](#),[ASPLOS'11](#)]
- Dune [[OSDI'12](#)]
- Mirage [[HotCloud'10](#), [ASPLOS'13](#)]
- NoHype [[ISCA'10](#)]
- Arrakis [[HotOS'13](#)]
- RaaS [[HotCloud'12](#),[CACM](#)]



- The cloud is a new kind of run-time environment
- An opportunity to rethink how we build system software
- We propose the **nonkernel**: a new kind of kernel
 - applications access their resources directly and securely
 - applications respond to changing resource costs
- The first nonkernel: **nom**

Thank you! Questions?

