A down-to-earth look at the cloud host OS

Malte Schwarzkopf

Steven Hand
“[...] In this position paper, we make a passionate and necessarily opinionated argument [...]”

“The abstract accurately describes this paper! It is passionate and opinionated and full of sensibilities.”
THE FOLLOWING PRESENTATION has been approved for SELECTED AUDIENCES ONLY by the AUTHORS.
Hypervisor

Linux kernel

Processes

"OS stuff"

interrupts, scheduling, preemption

User space threads

Language runtime

LL threads

Library code

User code

User code

Library code

LL threads

Language runtime

User space threads

Processes

Linux kernel

Hypervisor
Highly general

Familiar environment

Existing tools
Large base images

Booting...

Slow to spawn

Death by generality

Layering

[and yet, the programming models are often restrictive!]
All the other layers
What do we really need?
Batch

"Magic box"

i.e. some algorithm

Input data objects

Output data objects

...
Serving

Request handling logic
Virtualize custom μVMs

Back to the Eighties!
Numbers and experiment by Sören Bleikertz: [http://openfoo.org/blog/redis-native-xen.html](http://openfoo.org/blog/redis-native-xen.html)
Mantra:

Make the OS do exactly (and just) what is needed.
Execution control

Resource management

Isolation

Data access
Execution control

Non-preemptive scheduling

Dedicated cores

Centralize I/O mgmt

Statically link all user binaries
Resource Management & Isolation

Principle of OS buffer mgmt: request/commit

Request/commit interface

Backpressure for fairness

Embrace hardware heterogeneity
Resource Management & Isolation

AMD Opteron 6168

Intel i7-2600K
Benchmark your HW heterogeneity
Learn things about your architecture that you never knew!
http://fable.io
Data access

“Data object" abstraction

Global, deterministic naming

Transparent DO & buffer mgmt

[N.B. binaries are just DOs, too!]
Data access

Output UUID = task UUID | sequence num. | version | Input UUID$_{0,1,...,N}$

- Capabilities
- Consistency levels
I hear your cries...

“People should not need to know about OS-level stuff in order to program the cloud!”

“This is going to be a nightmare to program!”
Compiler support

New, bespoke toolchain

Simple interfaces

“Everything is a task”
Take-away:

How about we push the good things about MapReduce into the OS?
Take-away:

How about we restrict the OS to have simplicity and predictable performance?