

MicroswiftOS:

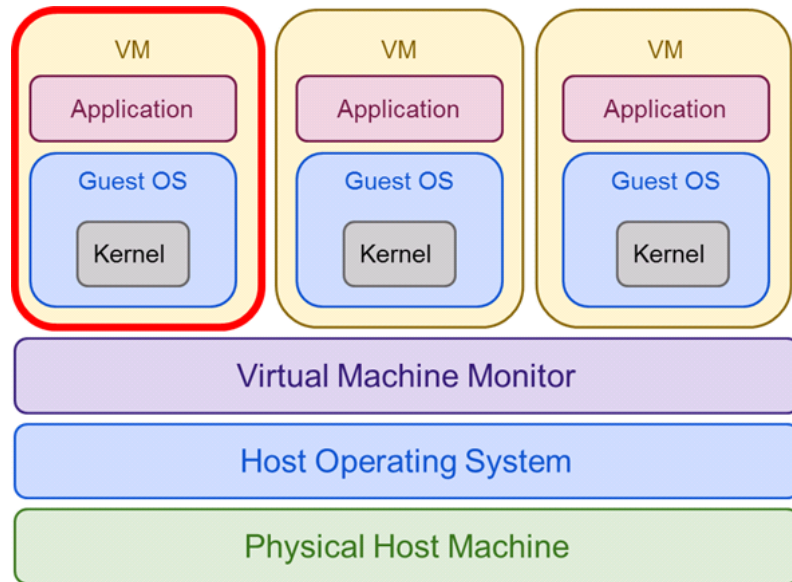
A Minimal OS for Cloud Computing

Yiran Menghui Chaowen Junhan

Motivations

Cloud apps are different!

- Dedicated VM
- Availability matters
- Unused OS features

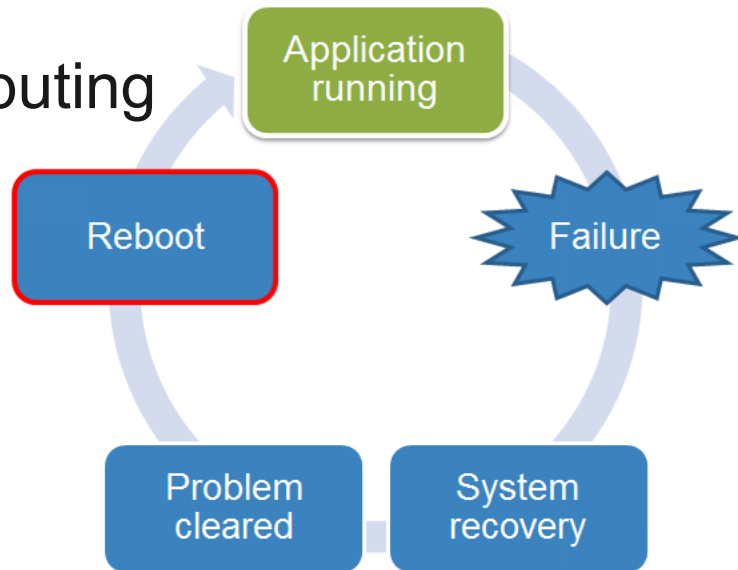


Goals

MicroswiftOS!

- Optimized for use in cloud computing

- ★ Minimal size
- ★ Faster boot-up



Kernel Optimizations

- Profiling User-mode Linux:
 - `perf`, `gprof`, FlameGraph
- Change Kernel configurations
 - Device drivers
 - Loadable module support
 - Kernel hacking
 - ...

System Optimizations

- Build a minimal software environment
 - Build from scratch. Install only necessary tools: glibc, binutils, gcc, important libraries, kernel headers, ...
- Change system configurations
 - Disable initramdisk
 - Disable fsck
 - Boot scripts

Evaluation

- MicroswiftOS vs. Arch Linux (baseline)
 - System completeness
 - System size
 - OS Boot-up time

Evaluation: Completeness

- Goal: to verify the correctness of MSOS
- Workloads:
 - Network service: tthttpd
 - Scientific computing:
 - max-flow solver
 - LP solver
 - linear equation system solver
 - I/O benchmark: inter-process communication

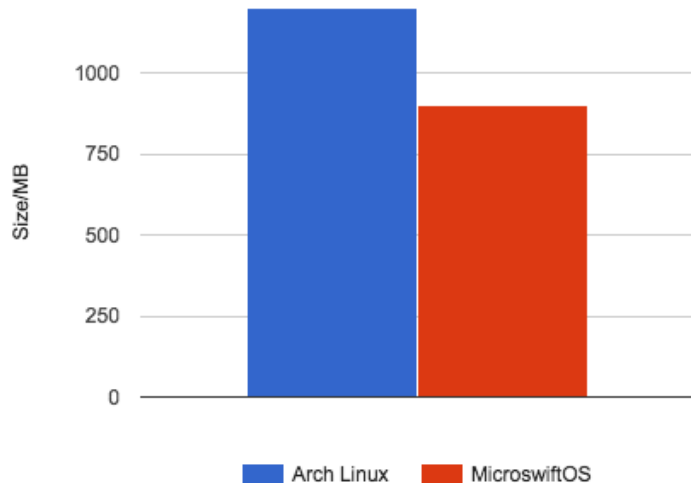
Evaluation: System Size

Operating system size after installation:

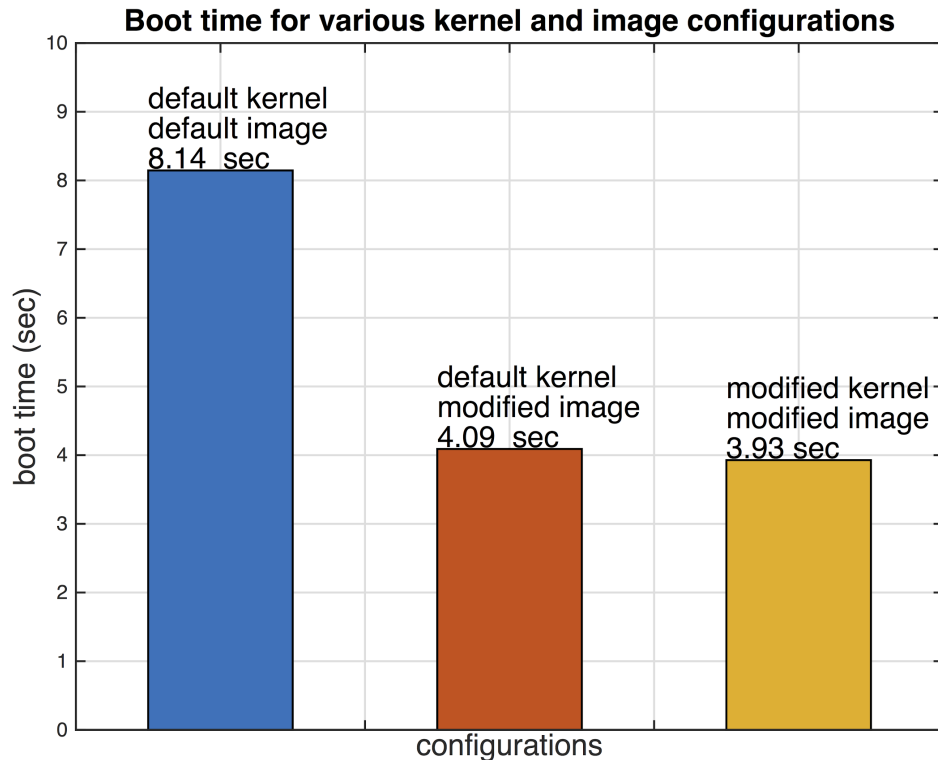
Arch Linux ~1200MB

MicroswiftOS ~900MB

~25% space savings



Evaluation: Boot-up Time



$$4.09/8.14=50.2\%$$

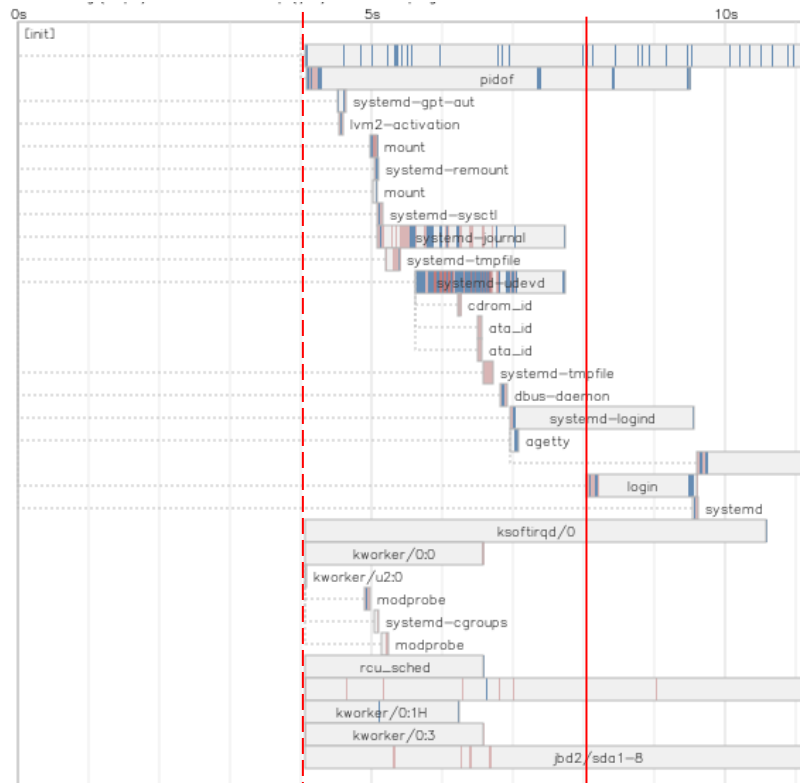
$$3.93/8.14=48.3\%$$

>50% time savings

Bootchart2 Analysis

Arch Linux

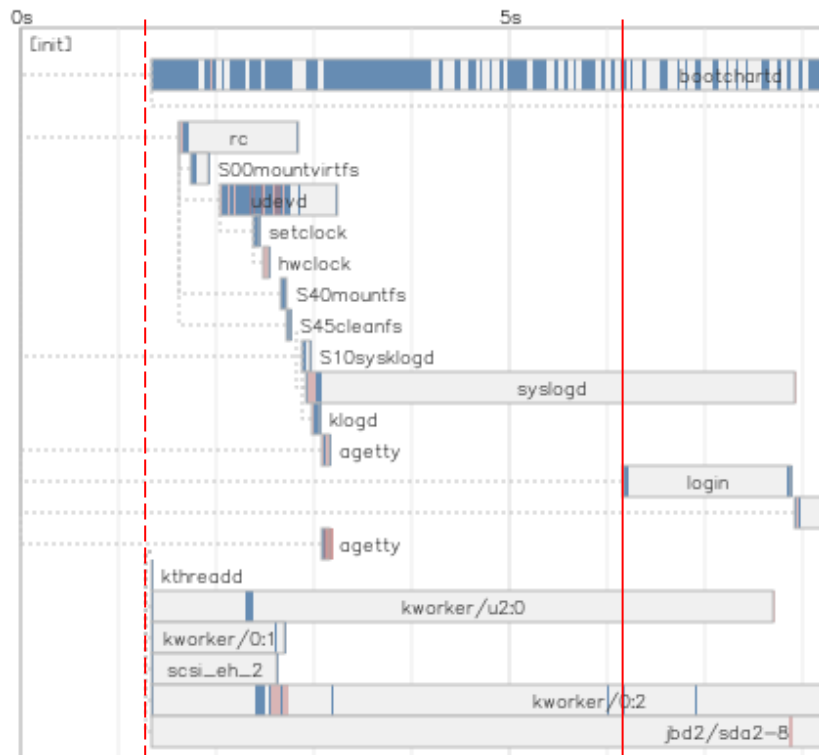
- Kernel booting
 - ~ 4.0 seconds
- Before login starts
 - ~ 8.0 seconds
 - 29 processes



Bootchart2 Analysis

MicroswiftOS

- Kernel booting
 - ~ 1.3 seconds
- Before login starts
 - ~ 6.2 seconds
 - 19 processes



Q & A

Thank you!