Manycores: challenges for OS

Flux Research Group
Anton Burtsev
Problems with current OS

- Solid monolithic structure
  - No explicit isolation and sharing
    - Hard to scale
- Single model of resource use
  - Unsuitable for specialized environments
    - Sheduling, security, etc. problems
    - Cannot run Linux and Windows in parallel
What we want?

- Reuse existing OS software stack
- Smaller, more flexible, more scalable
- Good balance between general and specific OS environments
- **Insight**: much like a data-center OS
  - Cloud of cores
How it will look?
How it will look?
How it will look?
What we need (in hardware)?

- Cross-core isolation
  - Protect cores from each other
- Core preemption
- Cross-core communication
  - Fast cross-core interrupts for communication
  - Message-like interface to send data without touching memory
Wild OS ideas

- Spatial scheduling
  - Application is always running
    - Just gets fewer or more cores
    - Hotplug CPU is already supported
- Specialized cores
  - One device driver per core
  - Language support for heterogeneity
Wild OS ideas (2)

- Thread farms
  - Libc calls out to another core
  - Exceptions, pagefaults - same

- Kilocores
  - Core becomes a resource
  - Pass core as an argument to a function
    - Needs hardware support for efficient address space transfer
Conclusions

- Ensure Linux' soft landing on manycores
- Evaluate how new OS structure affect performance
  - Identify anomalies
  - Figure out needed hardware support
Give us more cores!

Acknowledgments: Rob Ricci, Flux Research Group
Thank you.