Exploring Peer-to-Peer Virtualized Multithreaded Services

Kevin Vella and David Bailey

University of Malta

20/06/2011

CPA 2011 Fringe

Plan

- Overview of the experimental Xenos Framework
- Musings on multithreading, operating systems and virtualization

Virtual Machines and P2P nets

- Virtual machines provide flexible service hosting
  - Increased resource utilization
  - State isolation
  - Live migration

- P2P networks provide decentralised control
  - resilience, scalability and fault tolerance

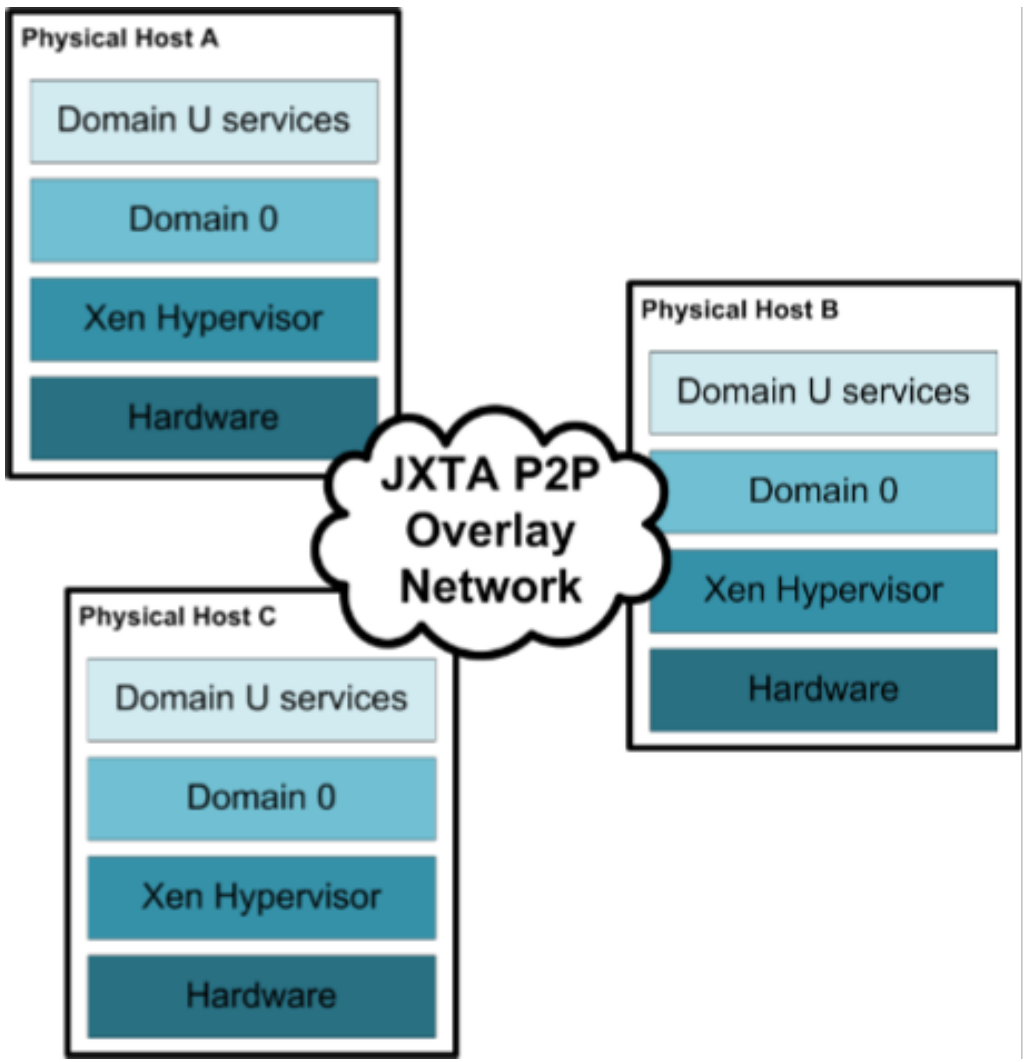
Xenos: A service-oriented, peer-to-peer framework for virtualized services

Xenos Objectives

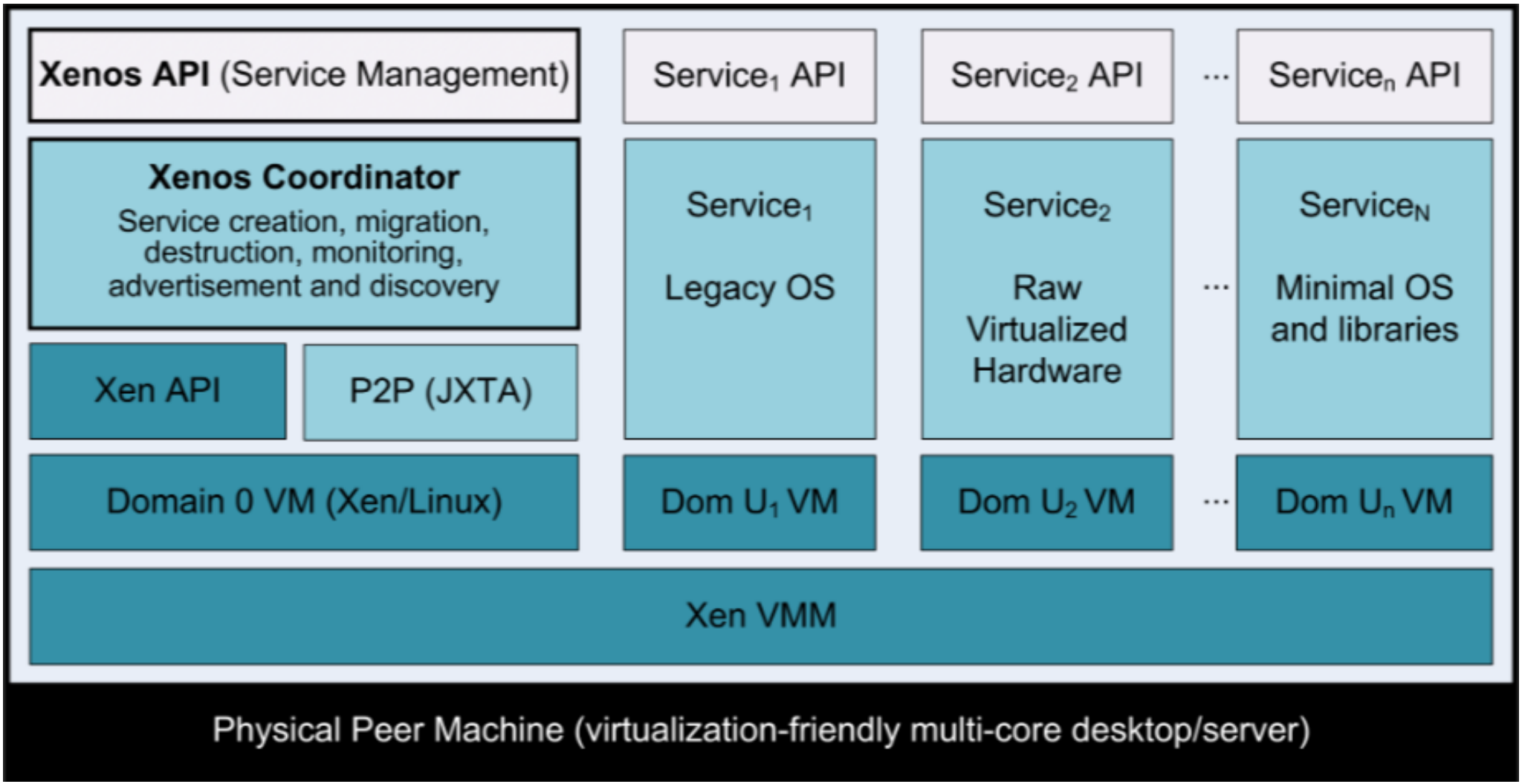
- A framework that allows for deploying and making use of services hosted inside Xen virtual machines
  - Web services running inside legacy operating systems
  - Custom services running within specialized execution environments (e.g. a stripped down Linux file system VM)
- Services are located through a JXTA P2P network as opposed to centralised service directories

Xenos

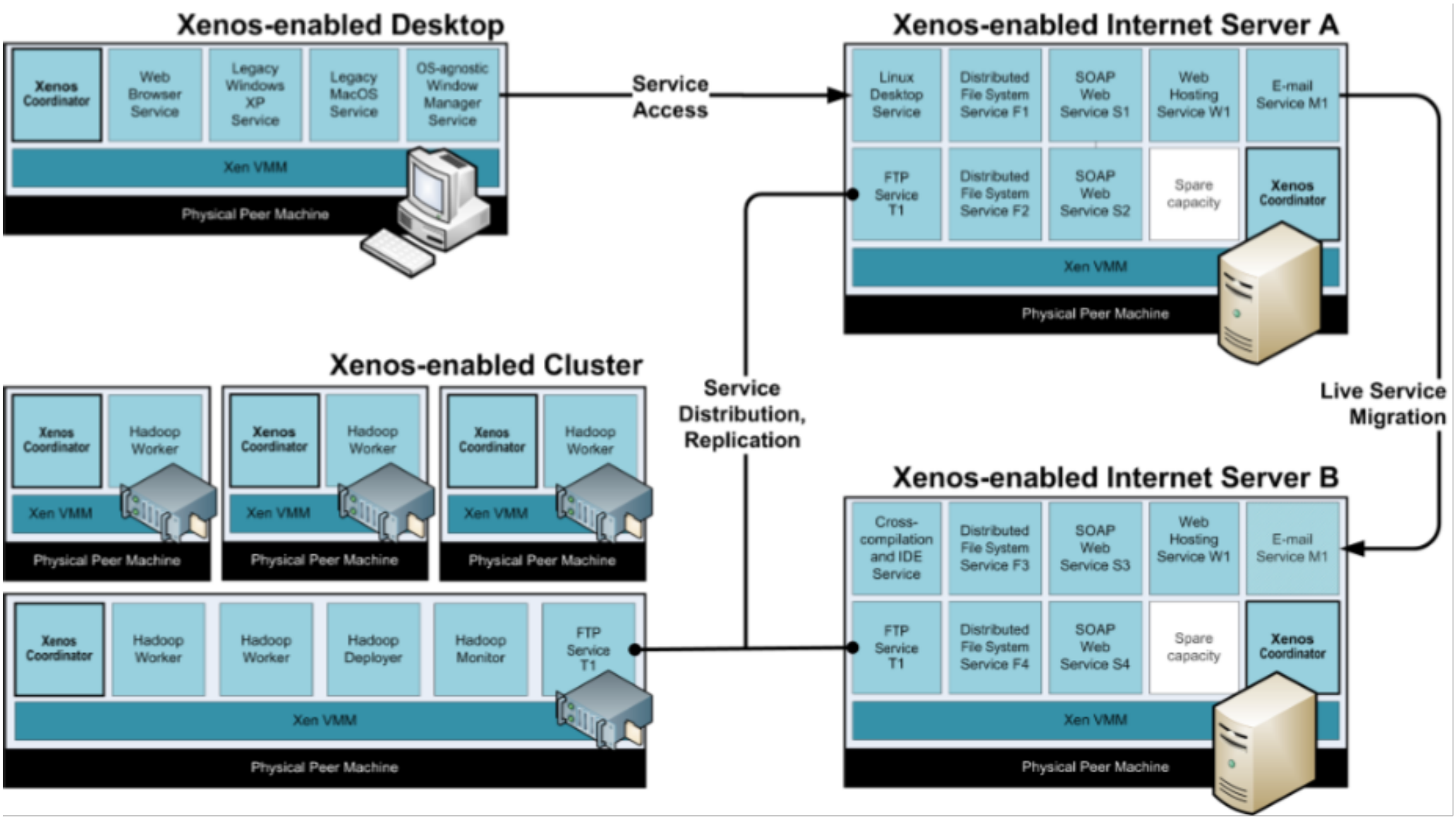
- **Xenos** is a Java application that runs within Domain 0 and connects to the JXTA P2P network
- It serves as the gateway through which services are hosted, discovered and delivered



Xenos Host Architecture

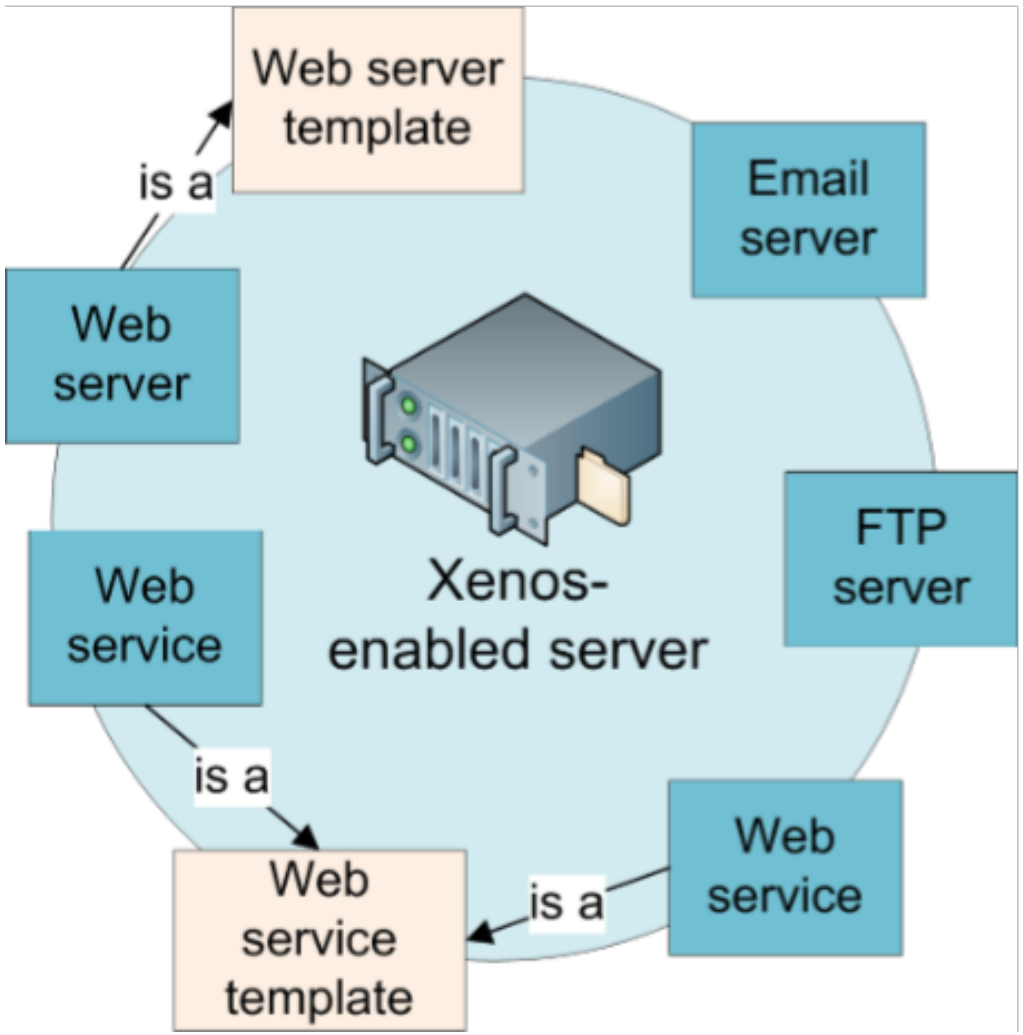


Physiology


Xenos-enabled Desktop
Xenos Coordinator
Web Browser Service
Legacy Windows XP Service
Legacy MacOS Service
OS-agnostic Window Manager Service
Xen VMM
Physical Peer Machine
Service Access
Xenos-enabled Internet Server A
Linux Desktop Service
Distributed File System Service F1
SOAP Web Service S1
Web Hosting Service W1
E-mail Service M1
FTP Service T1
Distributed File System Service F2
SOAP Web Service S2
Spare capacity
Xenos Coordinator
Xen VMM
Physical Peer Machine
Xenos-enabled Cluster
Xenos Coordinator
Hadoop Worker
Xen VMM
Physical Peer Machine
Xenos Coordinator
Hadoop Worker
Xen VMM
Physical Peer Machine
Xenos Coordinator
Hadoop Worker
Xen VMM
Physical Peer Machine
Xenos Coordinator
Hadoop Worker
Hadoop Worker
Hadoop Deployer
Hadoop Monitor
FTP Service T1
Xen VMM
Physical Peer Machine
Service Distribution, Replication
Live Service Migration
Xenos-enabled Internet Server B
Cross-compilation and IDE Service
Distributed File System Service F3
SOAP Web Service S3
Web Hosting Service W1
E-mail Service M1
FTP Service T1
Distributed File System Service F4
SOAP Web Service S4
Spare capacity
Xenos Coordinator
Xen VMM
Physical Peer Machine

Templates and Services

- Administrators can provide template domains from which service instances can be replicated (service factories)
- Template and service instance meta-data is advertised on the JXTA network

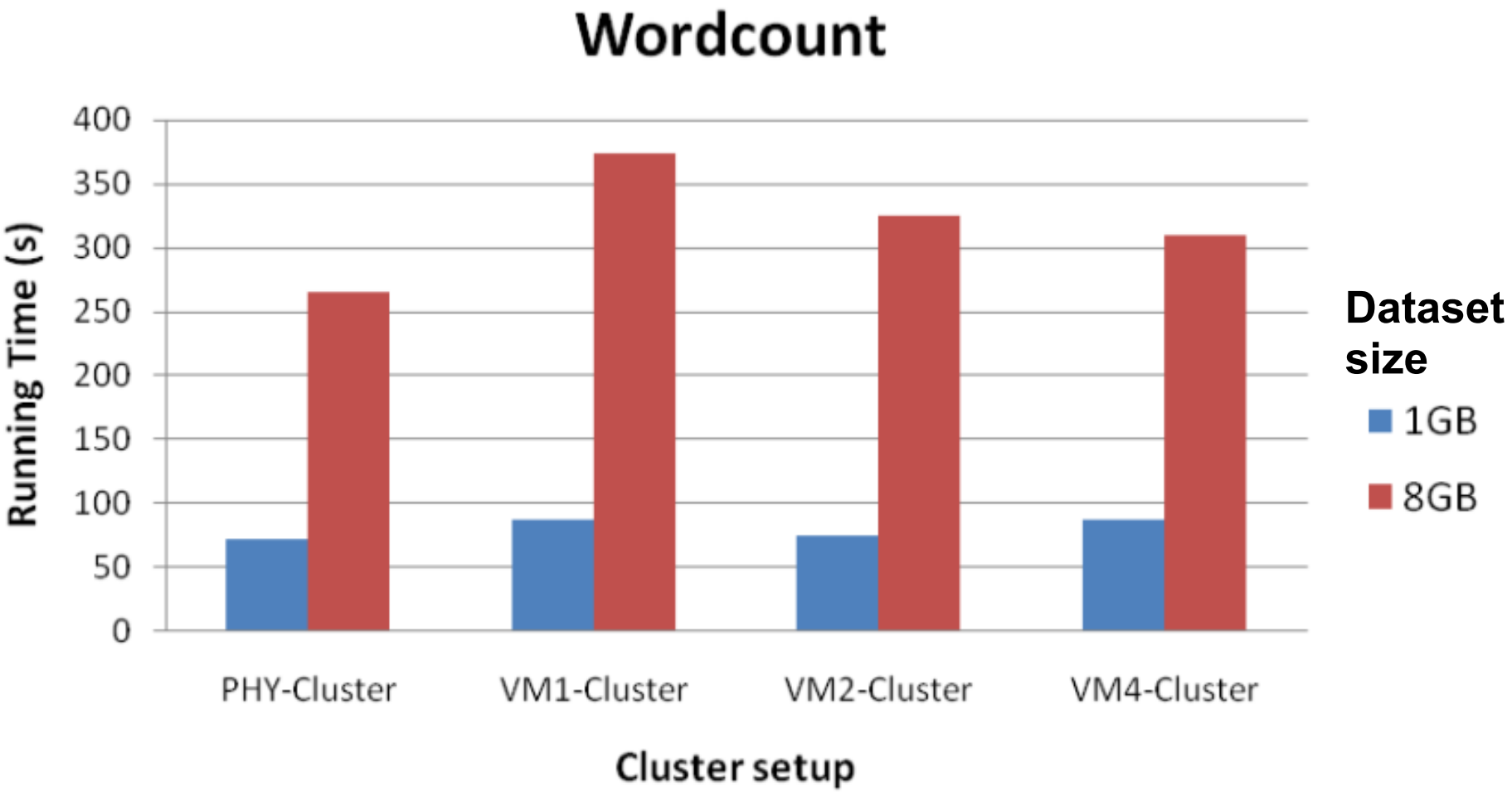


The Xenos API

- The search facility allows the caller to locate templates or service instances
- Services running on a host can be remotely controlled through the Xenos API
- A set of metrics is exposed for each service
- Services can be replicated dynamically by indicating the template from which to replicate
- Services can also be migrated on demand

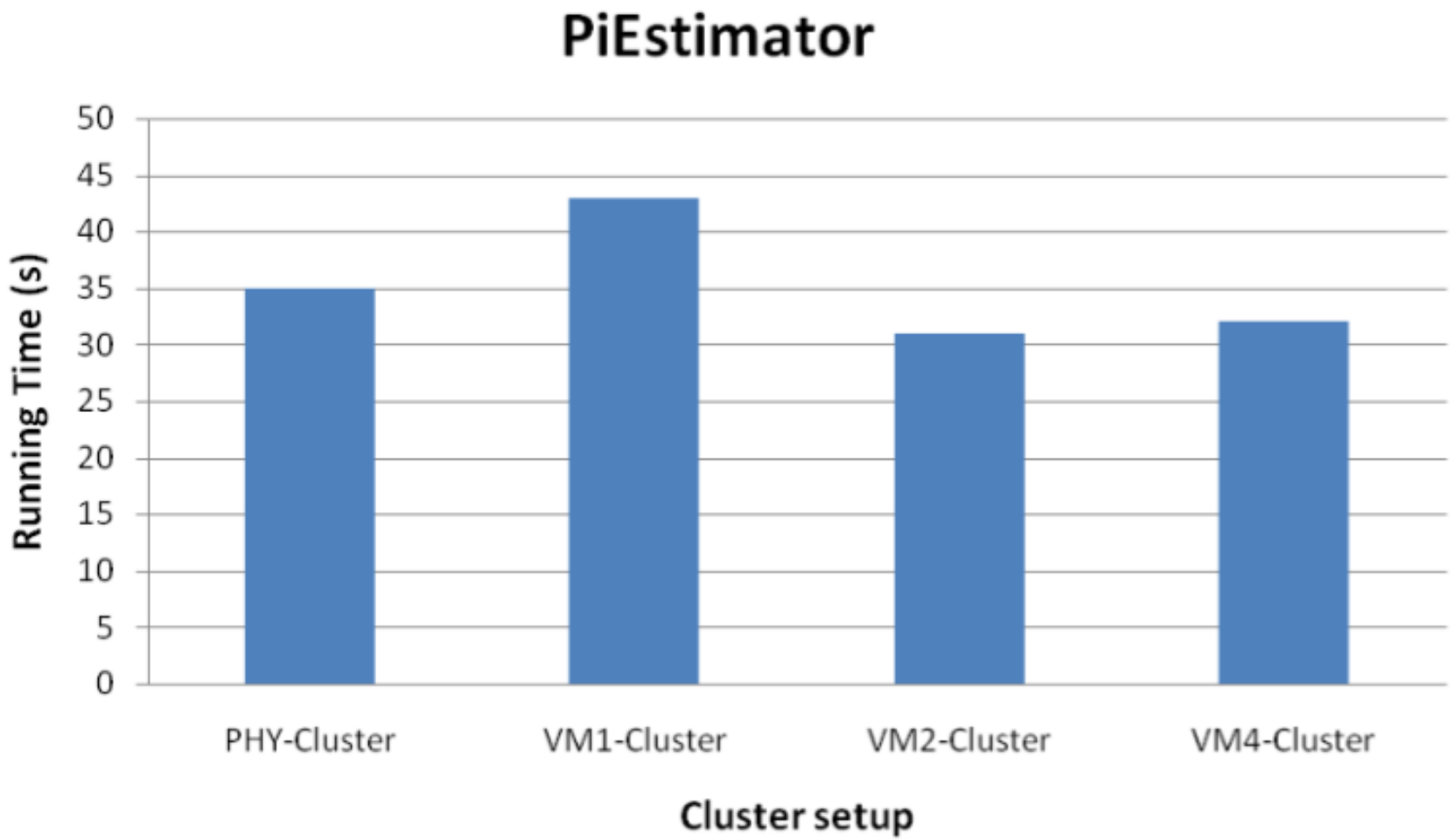
Hadoop job performance

- Wordcount



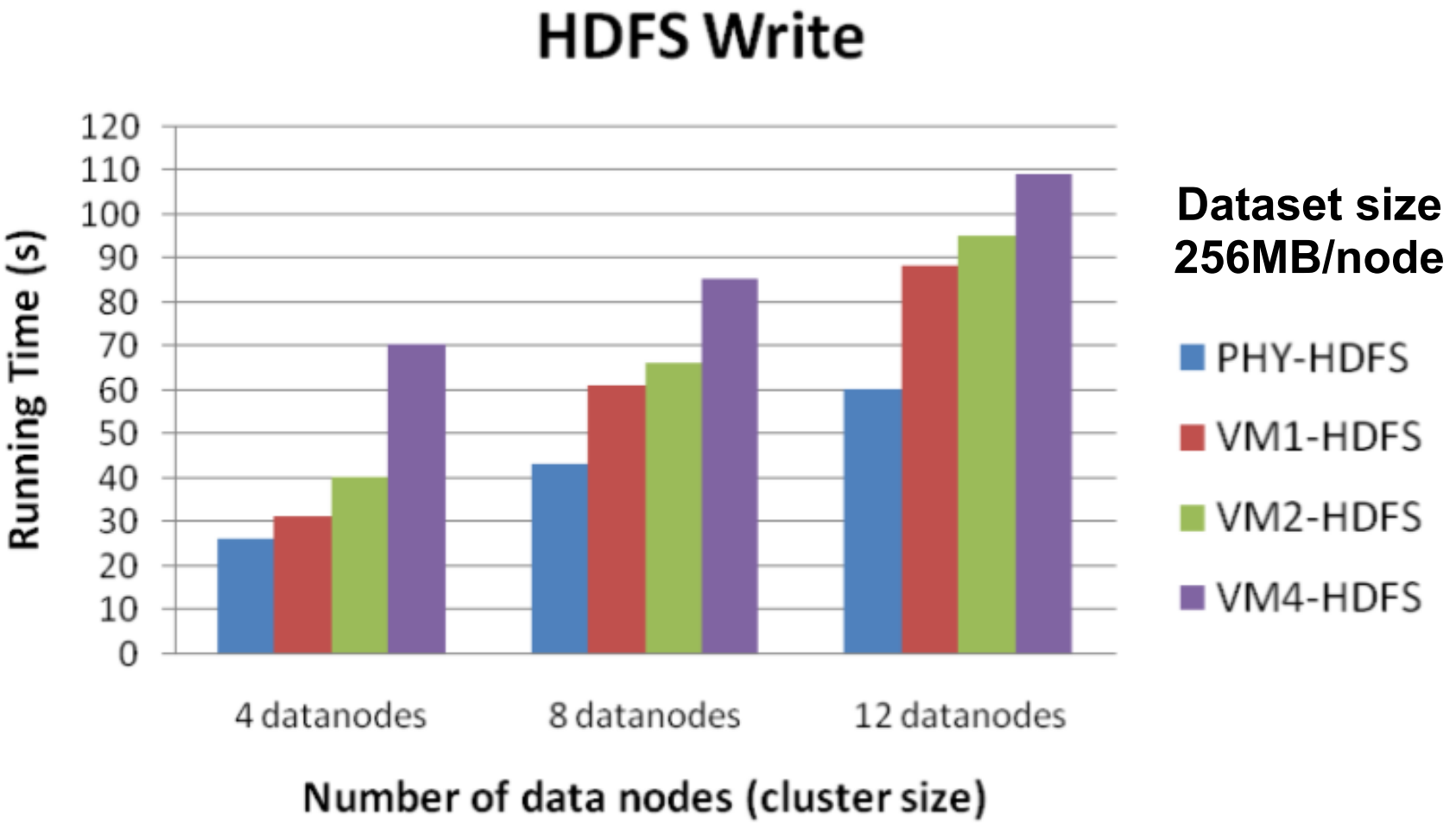
Hadoop job performance

- PiEstimator



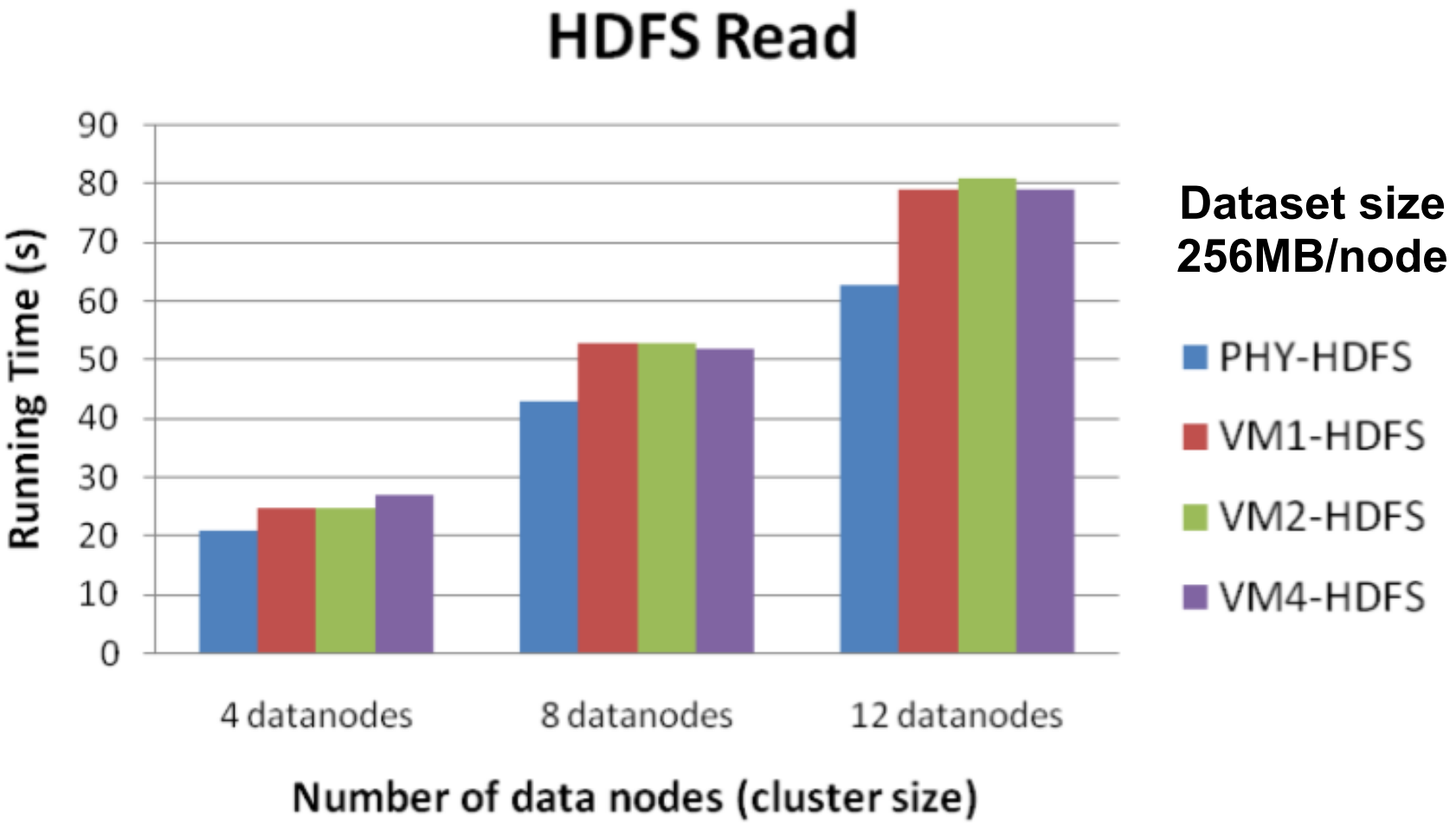
HDFS performance

- Varying virtualized nodes per host machine



HDFS performance

- Varying virtualized nodes per host machine



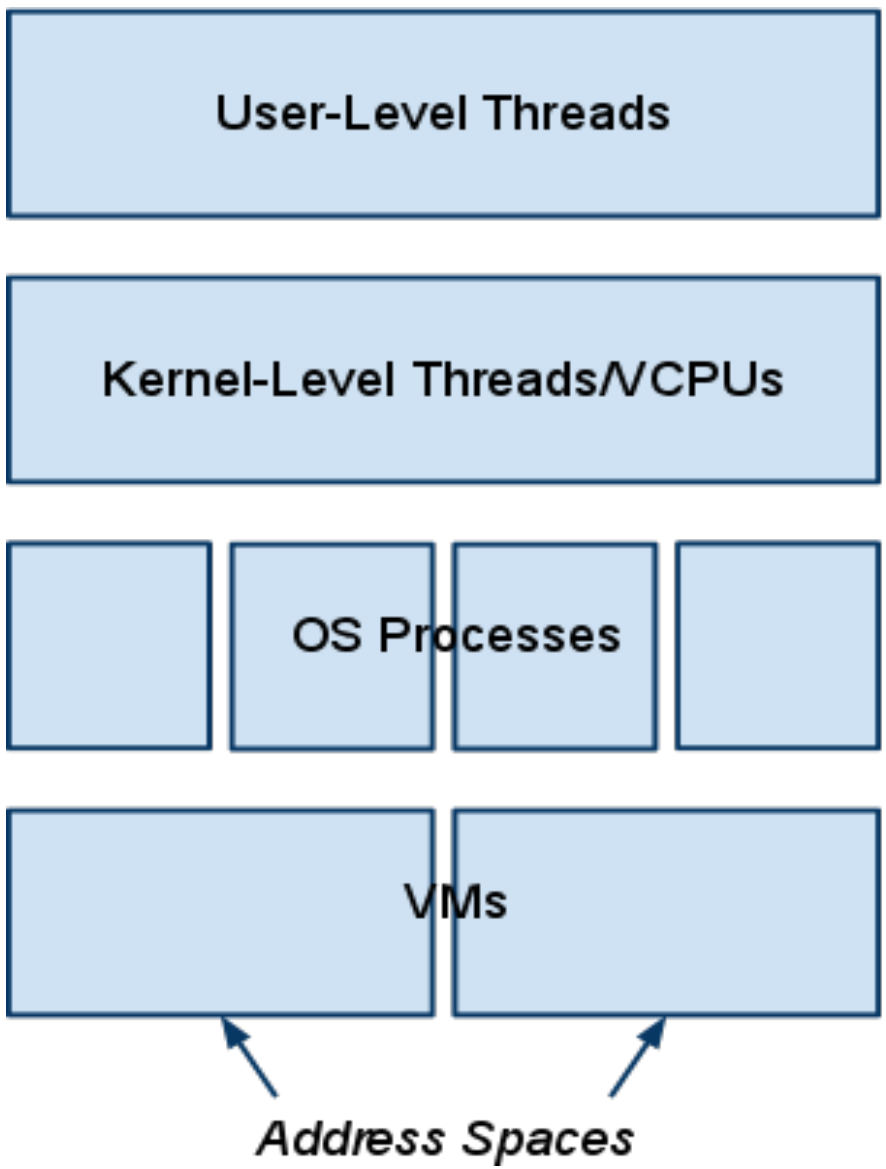
Observations

- I/O-intensive jobs suffer a penalty due to the degraded performance of the virtualized HDFS
- Processor-intensive jobs enjoy a performance benefit due to the increased availability of (virtualized) processing resources
- Our results are mostly consistent with 3rd party evaluations
- Xenos itself introduces negligible overhead and does not affect service performance

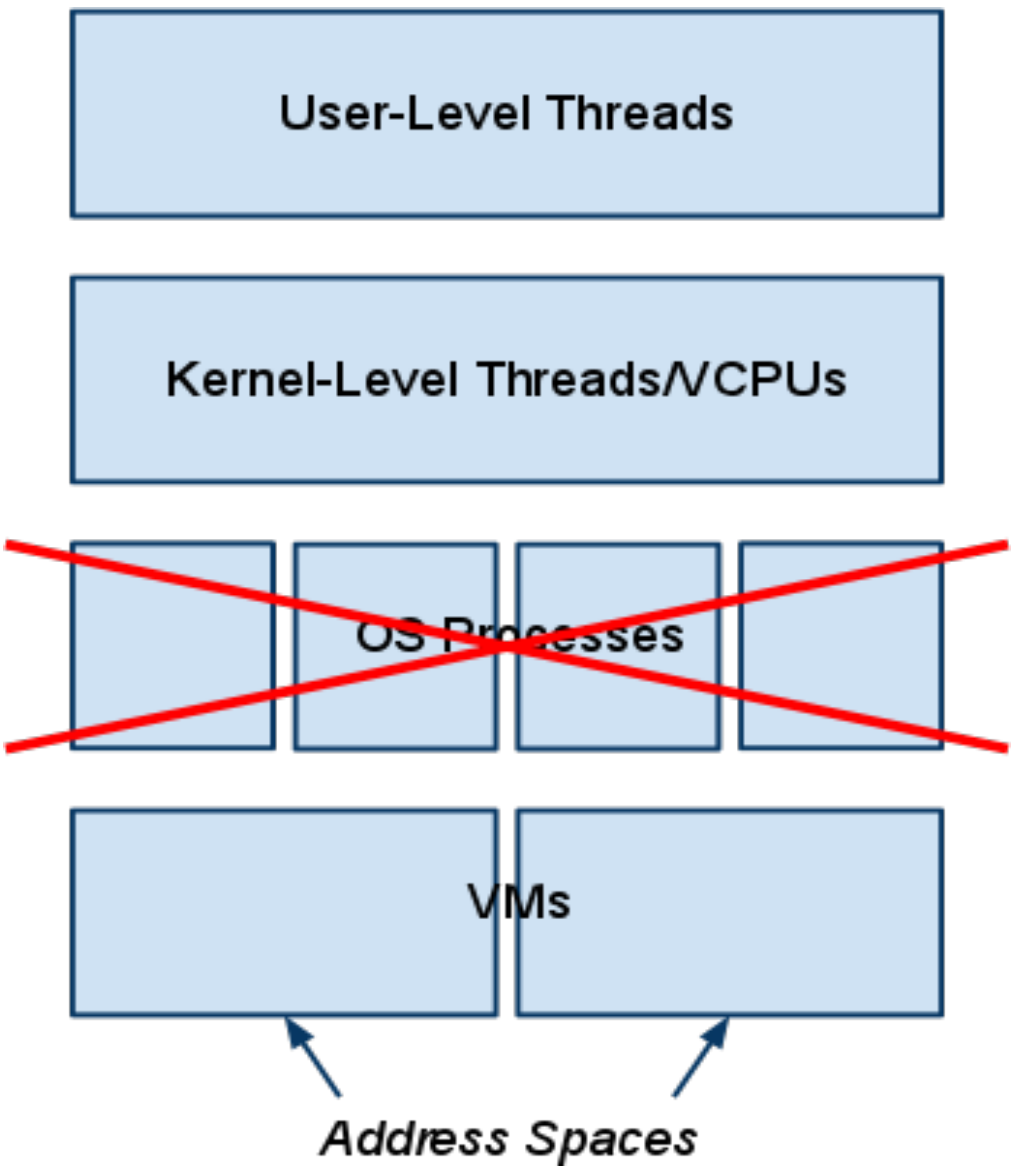
Multithreading and Virtualization

CPA 2011 Fringe

VMs, OS Processes and Threads



VMs, OS Processes and Threads

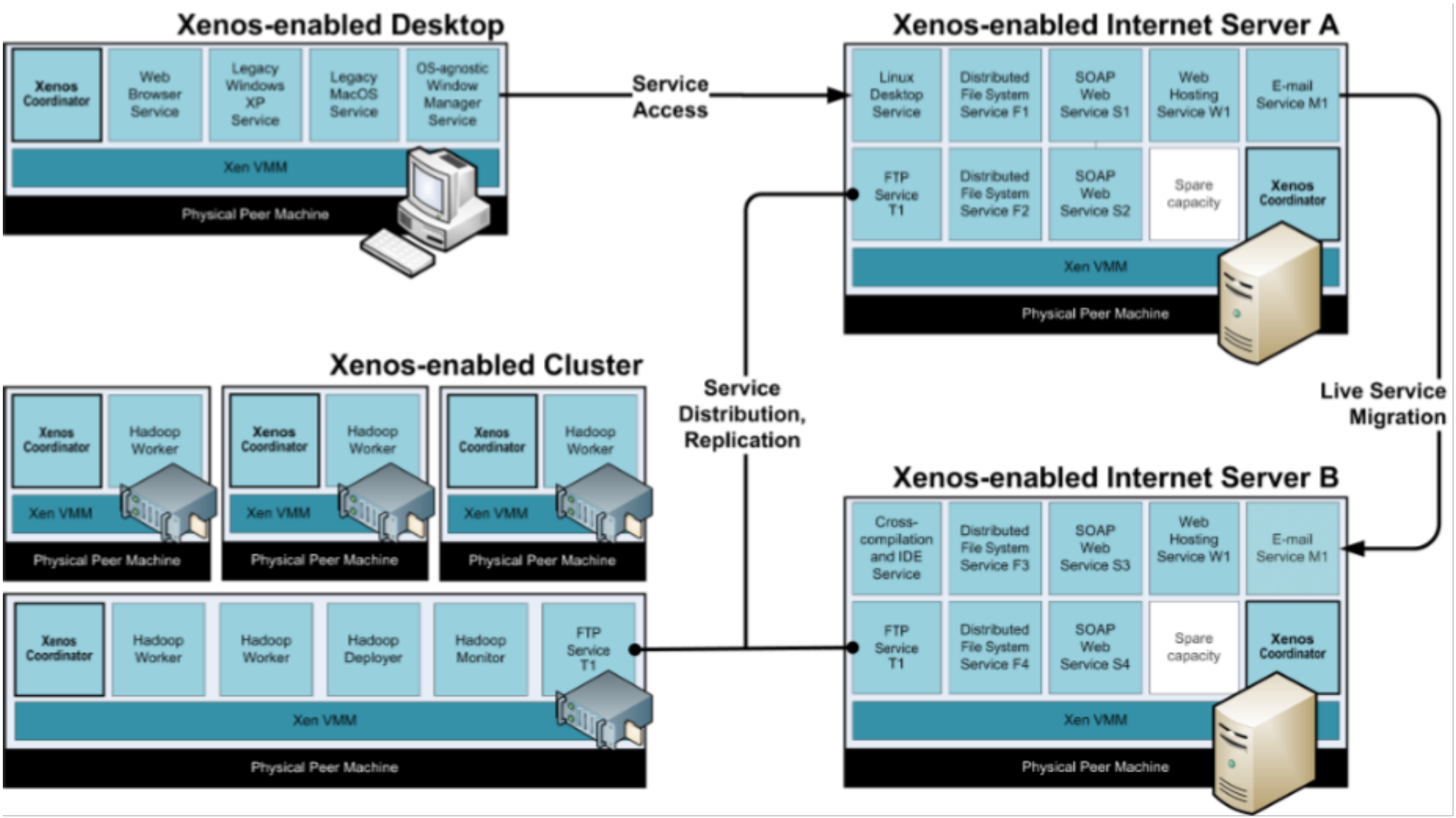


VMs displace OS Processes

- The virtual machine: a pragmatic vehicle for the distributed microkernel

- Example VMs (services)
  - **Virtualised bare metal thread schedulers** (*a la* RMoX)
  - File system (e.g. a stripped down Linux)
  - Legacy OS containers
  - VM managers (e.g. replication and fault tolerance)

The OS-agnostic OS



Thread Migration

- Intra-VM thread migration
  - Across VCPUs mapped onto physical CPUs sharing a physical address space
  - Using thread batching to improve cache utilisation
- Inter-VM thread migration
  - Across VMs sharing a physical address space
  - Across VMs on different physical address spaces
- Issues with global memory allocation when migrating threads across VMs

Channels

- Intra-VM channels
  - Low overhead wait-free channel implementation using shared memory

- Inter-VM channels
  - Negotiate cross-VM shared memory across VMs that share a single physical address space
  - Virtualised network device access for VMs on different physical address spaces
    - Multiplexed virtualization-aware devices and the I/O Memory Management Unit

Final Thoughts

- Prefer space sharing over time sharing to reduce world switch frequency
- Exploit emerging virtualisation-aware I/O devices to improve I/O performance
- Fluid thread migration across virtualised clusters
  - VM merging: migrate threads spread across several VMs onto one VM
  - VM splitting: migrate some threads to a new VM
- Hosted on-demand virtualised clusters
  - Data centre architecture vs. cluster architecture