

# Virtualization: Implications and Aspirations for Middleware

Industry Track Panel  
Middleware Conference  
4 December 2008

# “Obligatory” Trends Slide



virtualization.

Search Trends

Tip: Use commas to compare multiple search terms.

Searches [Websites](#)

Scale is based on the average worldwide traffic of [virtualization](#) in all years. [Learn more](#)

virtualization  1.00



# Buzzword-competitive



virtualization, grid computing, cloud computing, uti

Search Trends

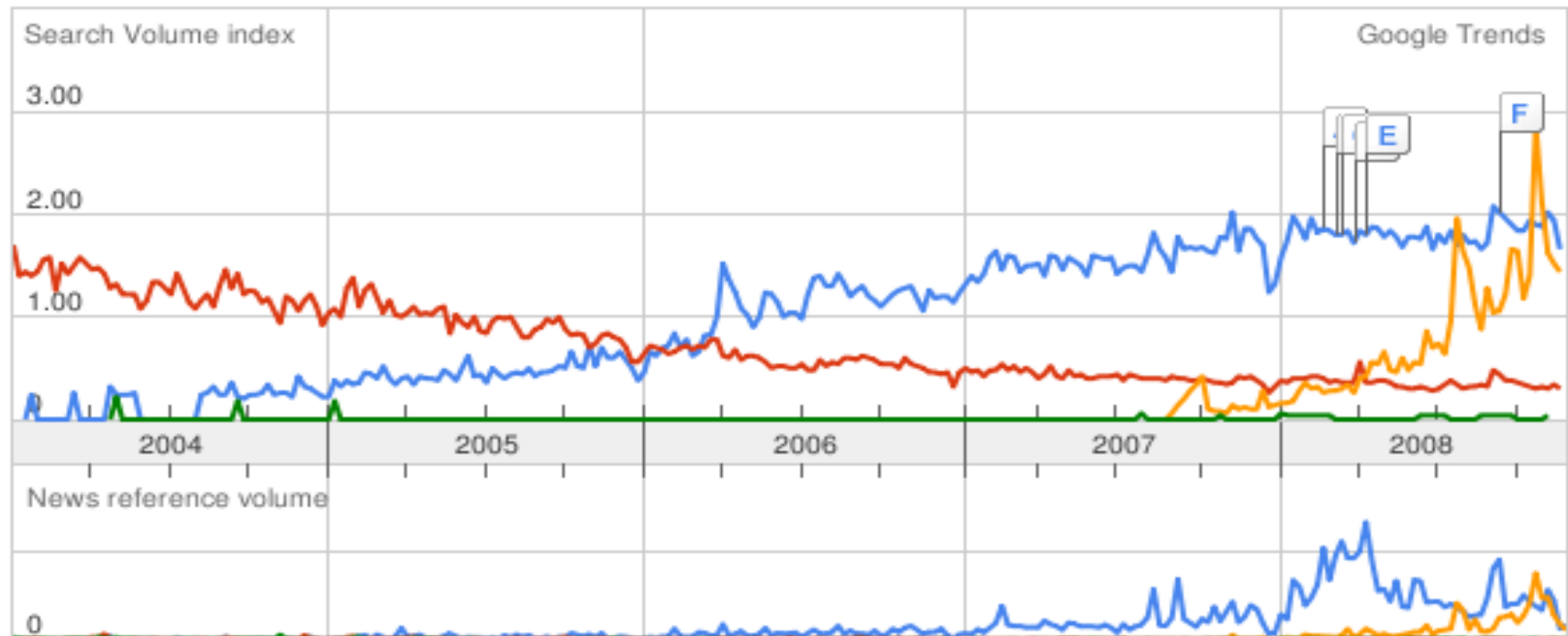
Tip: Use commas to compare multiple search terms.

Searches Websites

Scale is based on the average worldwide traffic of **virtualization** in all years. [Learn more](#)

**virtualization** 1.00 **grid computing** 0.72 **cloud computing** 0.16

**utility computing** 0



# Who We Are



**Fred Douglass**

IBM T.J. Watson Research Center

Current interests: Stream Computing

Related background: Operating Systems, File systems, mobility

# Who We Are



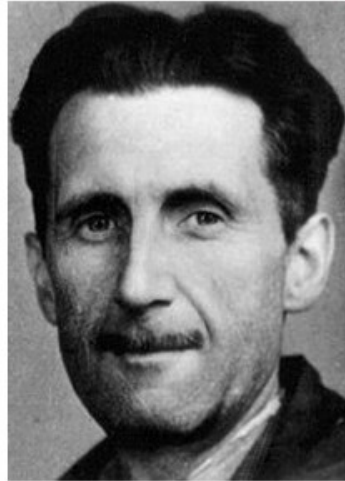
Professor **Roy Campbell**

University of Illinois at Urbana-Champaign

Current interests: Cloud Computing

Related background: Operating Systems, Security, CyberInfrastructure

# Who We Are



**Ernie Cohen** (looks like George Orwell)

Verification architect for the Microsoft Hypervisor (core of Hyper-V)

Current interests: verification of system software

Related background: security, distributed systems, middleware

# Who We Are



**Prof. Atul Prakash**

University of Michigan

Current interests: security and privacy, middleware for emerging applications

Related background: cloning file system for virtual servers (Prism),  
using virtual machines to detect malware

# Who We Are



Prof. **Liuba Shrira**

Professor at Brandeis University, Affiliated Researcher at CSAIL/MIT

Current interests: time travel in storage, security

Related background: storage replication, software upgrades,  
spelunking (kind of time travel)

)

# Who We Are



**Professor Roy Campbell**  
University of Illinois at Urbana-Champaign  
Current interests: Cloud Computing  
Related background: Operating Systems, Security, CyberInfrastructure



**Fred Douglass**  
IBM T.J. Watson Research Center  
Current interests: Stream Computing  
Related background: Operating Systems, File systems, mobility



**Ernie Cohen** (looks like George Orwell)  
Verification architect for the Microsoft Hypervisor (core of Hyper-V)  
Current interests: verification of system software  
Related background: security, distributed systems, middleware



**Prof. Atul Prakash**  
University of Michigan  
Current interests: security and privacy, middleware for emerging applications  
Related background: cloning file system for virtual servers (Prism), using virtual machines to detect malware



**Prof. Liuba Shrira**  
Professor at Brandeis University, Affiliated Researcher at CSAIL/MIT  
Current interests: time travel in storage, security  
Related background: storage replication, software upgrades, spelunking (kind of time travel)

# Themes & Questions

- What are we gaining with virtualization?  
What are we sacrificing?
- What new functionality is enabled?
- Where will virtualization be in 1,5,10 years
- What advice would you give Vmware, Xen, etc? What are we missing?

# Panel Format

- Each panelist has about 8 minutes to cover these topics
- This leaves close to  $\frac{1}{2}$  hour for questions
  - Perhaps **one** question after each panelist finishes

# Virtualization: Implications and Aspirations for Middleware

Roy Campbell

University of Illinois at Urbana-Champaign



# Questions



- What do we gain from virtualization?
  - Flexible growth and reuse of systems and configurations. Cost sharing. Reliability and availability. Shared improvements and innovations
- What do we lose?
  - Simplicity, supply side independence, common mode failure
- What key features are missing?
  - Cross cutting performance guarantees
  - Security, privacy

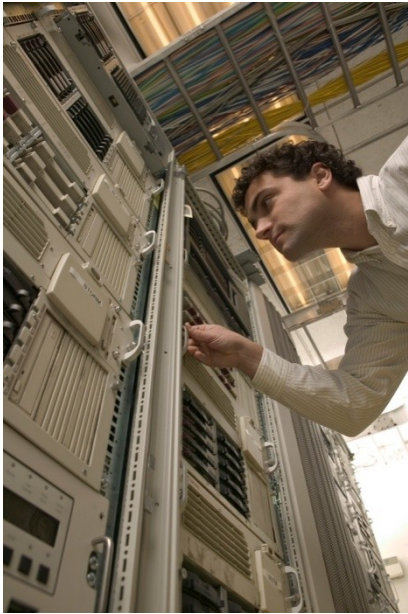
# Outsourcing the Academic Infrastructure

- Virtualized resources:
  - Research: shared huge data sets, collections, scalable processing, multiprocessing
  - Education: highly available, communication, instructional resources (email, calendar, courses,...)



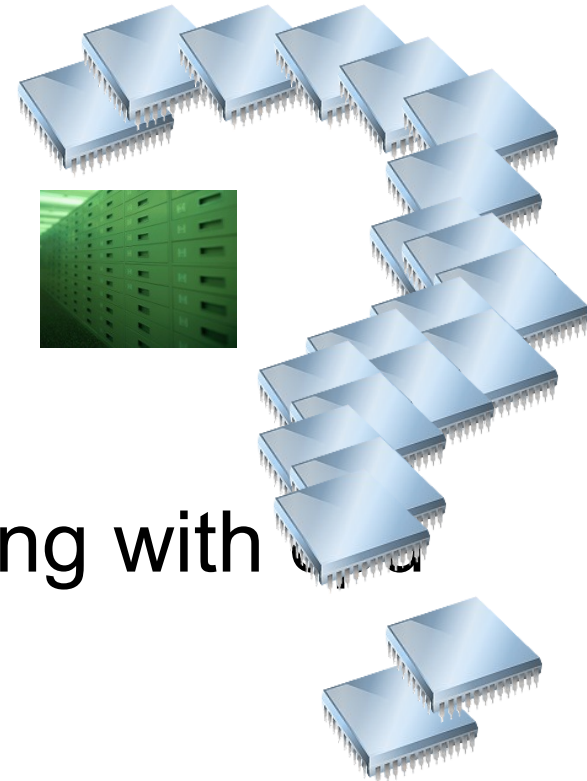
# Architectures to support virtual provisioning

- Vlans, vwans, virtual OS, virtual storage
- Location modularities: chip, rack, cluster, room, city, network, provider
- Declarative (graphic) languages for virtual provisioning



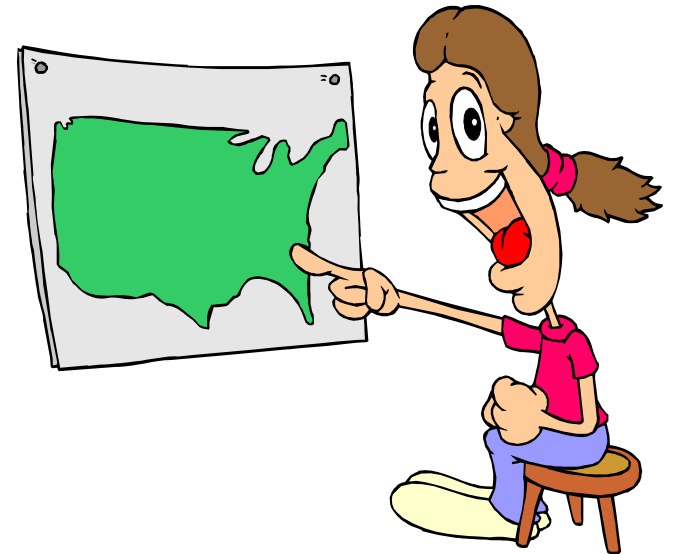
# Storage/scheduling virtualized data intensive systems?

- Moving computation or data?
- Network infrastructure
- Disks/storage per node
- CPUs per node
- Can one mix storage scheduling with scheduling



# Geographic location and virtual provisioning

- how do you take into account geographic location cleanly in virtual provisioning
- archival, fault-tolerance, latency, data gathering, users





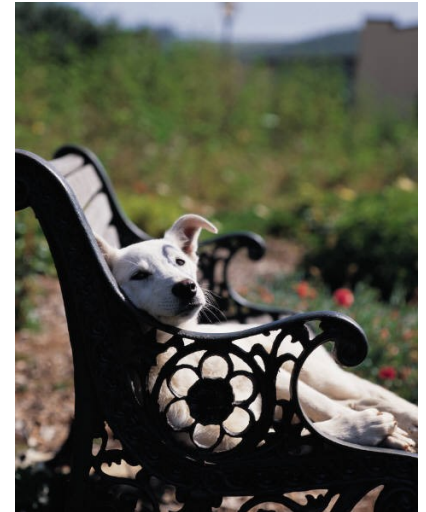
# Limits of virtual provisioning

- data intensive computing  $\neq$  supercomputing
- Can virtual provisioning provide both?
- How are cross-level performance attributes offered?



# Fair benchmarks for virtualized provisioning/virtual systems

- Availability/performance/reliability....
- Data intensive
- Cloud for Casual Users
- Supercomputing – parallelism?
- Network intensive?



# Guarantees versus off peak earnings

- virtual provisioning can guarantee a large corporate its essential on-demand operation peaks
- Sell surplus capacity as cloud provisioning





# Trading and selling provisioning?

- Should there be an exchange market for trading and selling virtual provisioning?
- Large finance companies, search companies...
- Peaks and troughs

# High end network fabrics with intense computation

- Virtualized interactive 3D (optimistically Holographic) video. High end network fabrics with graphic processing studios between remote sites) and 3D/holographic displays



# Where will virtualization be in 1,5,10 years

- Mixing data intensive, supercomputing and on demand clouds
- End to end virtualization
- Multi-core virtualization
- Cellular access
- Instant “on”



ernie cohen

# why processor virtualization?

- business driver: flexible provisioning
- middleware driver: OS-neutral illities
- a chance to overcome the OS legacy
  - failure to achieve a process abstraction
  - OSs can't interoperate
  - OSs unworthy of trust
- VMs are the new processes
- hypervisors are the new OS kernels
  - potentially trustworthy (2-100 KLOC)
- middleware is the new modular OS

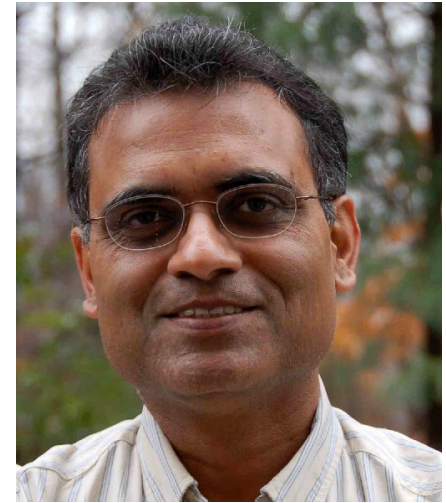
# the future

- in 5-10 years, all x64 computing will be hypervised
  - hypervisor-on-metal everywhere
  - microkerneled (no devices in the hypervisor)
  - <5% virtualization overhead
  - (mainly thanks to improved hardware)
- market consolidation
  - a few good, free commodity hypervisors
  - machine migration across vendors
- a rich virtualization ecosystem, with VMs as components
  - small, specialized OSs
  - virtual devices
  - illities
- a path to building trustworthy systems
  - verified hypervisors
  - verification tools for low-level code

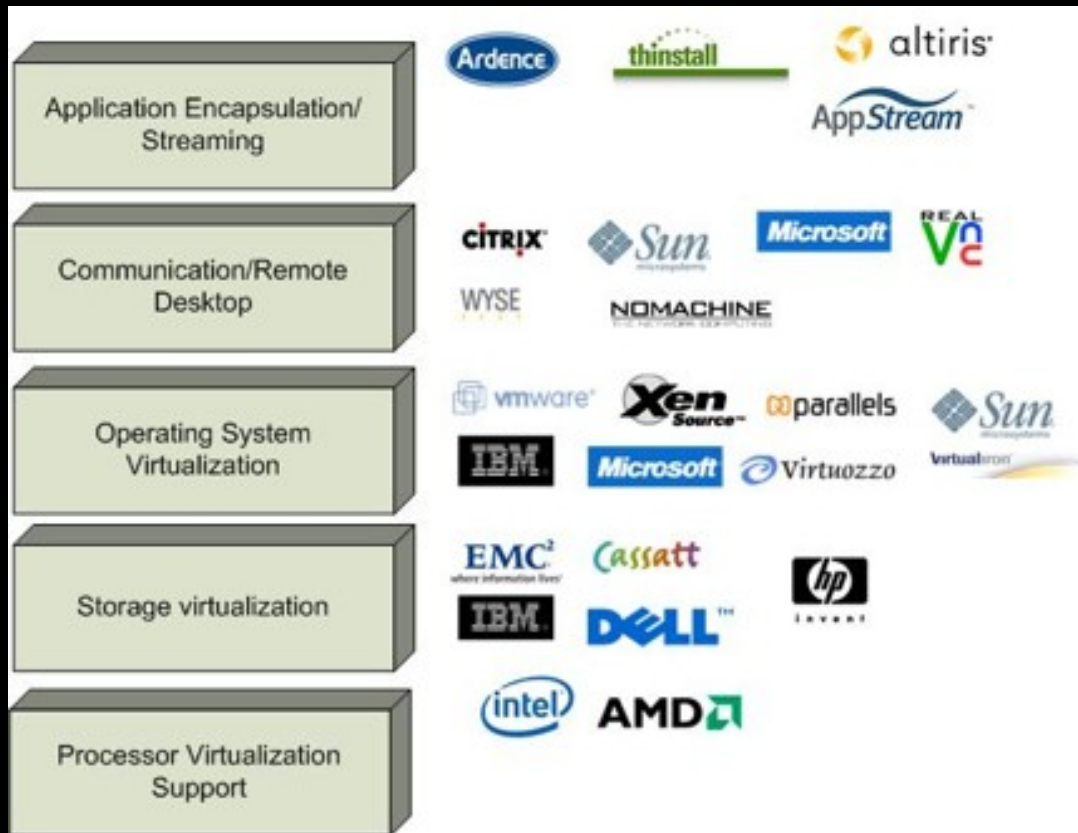
# what do we need from the hardware/hypervisor vendors?

- a stable, well-defined platform to build on
  - formal specs
  - strong enough for legacy OSs
  - flexible enough for future hardware
  - buyin from hardware vendors
  - consolidated hypercall API (POSIX for VMs)
- don't repeat the mistakes of the OS world
  - e.g. don't try to force the community to sit on top of your favorite new management abstractions
- make hypervisors a free commodity; make your money in the ecosystem

# Atul's Slides



# Current Promises

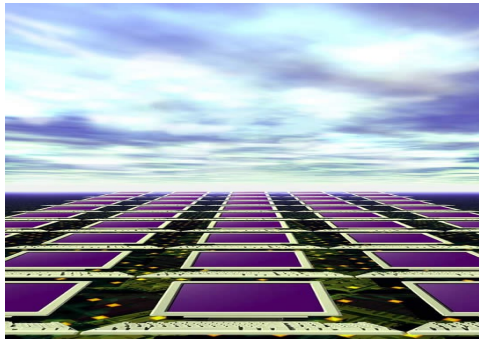


# What is missing?

- Maintaining virtual machines: cost of software maintenance cost of virtual PCs not lower: currently, maintained like regular machines
- Dormant virtual machines can pop up and mess up security
- Easier to "steal" the entire virtual PC
- 3D-graphics and I/O passthrough (some hardware support exists from Intel, but software support lagging)

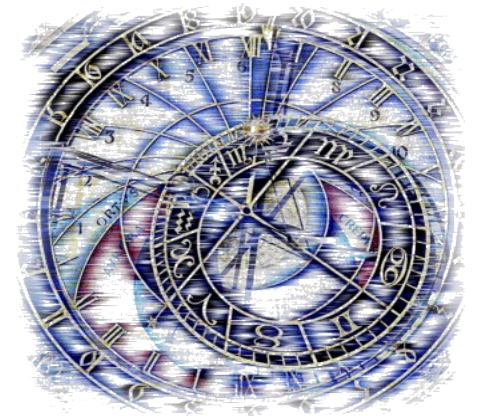
# 5-10 years from now?

- Rethink storage and file systems to support virtualized environments.
  - Want features of virtualization (snapshots, migration etc.), but security  $\geq$  standard file systems
- Dormant VM security
  - Patches without requiring online VMs?
  - VM identity management?
- Dynamic graphics and device pass-through



# Time travel in the Virtualized Past

Liuba Shrira  
Brandeis University



# Time travel is interesting!

"The further back you look the further into the future you see" Churchill

Have you ever wondered: have I seen this before, what else happened then, what is different now?

Desktop

Apple's Time machine/Time capsule .. short trips

Storage

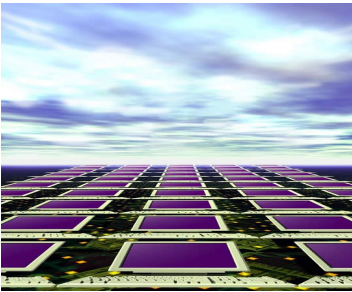
Elephant .. long trips, my area

Literature

Pensieve from Harry Potter

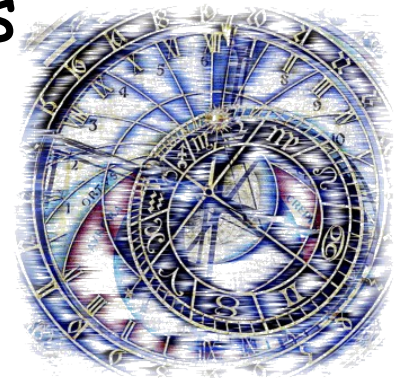


# Revelation: Time Travel is Virtualization



Much virtualization concerns space -  
software/hardware layers, platforms,  
resources, geographical locations

Virtualization in time is  
all that and more..



# What does it take to TT ?

- Preserve past unobtrusively
  - the stored past can look nothing like the current state
- Reconstitute as needed...
  - so *old* unmodified apps can run on it
  - hence virtualization
  - but also *new* apps
  - not just in pre-planned



"The past is a foreign country,  
they do things different there..  
L.P.Hartley"

- Software evolves
- Hardware gets obsolete
- Privacy norms wane and wax ...

# Technical challenges, if you were a graduate student looking for a topic..

- **Virtualization** problems:
  - at what level to capture past states: application, files/DB, storage, VM, or processor - intel 86 - the slowest evolving part!
- **Time travel** problems:
  - how to create past unobtrusively?
  - how far back and how fast you want to travel?
  - on budget? : some states more interesting



Your challenge is a new kind of **virtualization problem**: how to organize your systems today to enable better time travel tomorrow

# And now to Fred's Questions

- What did I gain with virtualization?
  - Can not do much without it..
- What are we sacrificing?
  - More tradeoff than sacrifice:
    - More *efficient* to integrate past *inside* the virtualization level of present state. *Below* maybe *simpler*, *above* more *complex* (see our papers)
- What new functionality is enabled?
  - Can run *retrospective* computations, *adaptive past* for faster and slower travel, *GC*
- Where will virtualized past be in 1,5,10 years
  - In your brain eventually
- What advice would you give Vmware, Xen, etc?

# Open Discussion