OpenSolaris on z: user experiences and perspectives at Sun

Jeff Savit
Principal Field Technologist
Sun Microsystems

Agenda

- Background (I'll try to make it painless)
  - > Sun Microsystems
  - > what I do at Sun, which by funny coincidence is:
    - – Solaris
    - – Virtualization
- OpenSolaris on z (sirius)
  - > background
  - > user experience, performance
  - > perspectives

Overview of Sun

Global

- Business Presence in 160 Countries
- Fortune 187

Business Strength

- $13.8 billion annual revenue
- $5.19 billion cash*

Innovation

- 11,000+ patents
- $2 billion annual R&D

Communities

- 11 million Solaris licenses
- 6+ million Java developers
- 5.5+ billion Java devices

*As of Q1FY08

More Open Choices

Flexible and Heterogeneous with Zero Barrier to Exit

Application Infrastructure	Java · Project Glassfish · NetBeans · php · The Apache Software Foundation · JRuby
Database Platform	MySQL · Java DB · Apache Derby · PostgreSQL
Virtualization	Sun xVM · innotek VirtualBox · vmware
Operating System	redhat · suse · ubuntu · Microsoft Windows · solaris
Architecture	intel · AMD 64 Opteron · OpenSPARC

Platforms for the Web Economy

What I do

- I'm a “Principal Field Technologist” focusing on Solaris and virtualization (what I'll discuss in the following slides)
  - With customers, marketing, engineering
- Author of a Blueprint (like IBM's Redbooks). Surely the only one in Sun that mentions the MVMUA
  - http://www.sun.com/blueprints/0807/820-3023.html
  - Also a contributor to an IBM Redbook about WebSphere running on Solaris. Years as an IBM customer, and I finally contribute to a Redbook as a vendor!
- Mainframe topics are a sideline/hobby

A very brief overview of Solaris and OpenSolaris

Solaris – what is it

- Sun's implementation of Unix (duh...)
- Available in production on SPARC and x86
  - From laptops to supercomputers with 2TB RAM and more than 100 CPUs
  - Massive hardware compatibility list on wide range of vendors' systems
  - Massive ISV portfolio
- Many world records in performance
- Heavily used in government, telecom, manufacturing, web/web2.0, financial services, healthcare, pharma, education, etc

Solaris features beyond generic Unix – why needed

- 15 years ago, in my “Strategic Outlook for VM”, I complained that Unix didn't have:
  - > Resource management
  - > A security model besides God-like “root” vs. peon.
  - > RAS (reliability, availability, serviceability) for staying up under duress, diagnosis, software management
  - > Virtualization
  - > I complained about “vi” too
- **Solaris has all these features**
  - > Sometimes in forms you would recognize immediately
  - > Sometimes delivered in quite different ways
  - > “vi” is still there, but you don't *have* to use it :-)

Just a few of the Solaris features beyond generic Unix

- Resource management (CPU, RAM, swap, etc)
  - Solaris systems easily handle high production %utilization
- RAS services, including automation
  - Solaris systems running for many months without reboot
- Granular security model based on “least privileges”
- Built-in virtualization (Solaris Containers)
  - Many virtual Solaris instances on same server
- Advanced filesystem ZFS, with RAS, performance
  - Transactional I/O, snapshot/clone, no `fsck` **ever**
- Dynamic tracing (DTrace) of both user and kernel

What's New with Solaris?

IBM jumps on the Solaris bandwagon

vnunet.com

Sun to Acquire MySQL AB, increase investments in PostgreSQL, Apache Derby

Sun/Dell pact expands hardware choices for Solaris

GCN Government Computer News

Sun Solaris going on Fujitsu's Intel servers

InfoWorld

Sun Microsystems to Acquire Innotek

AP Associated Press

Ian Murdock leaves Linux Foundation, joins Sun

linux-Watch eWEEK

Sun, IBM, Sine Nomine demonstrate Solaris on mainframes

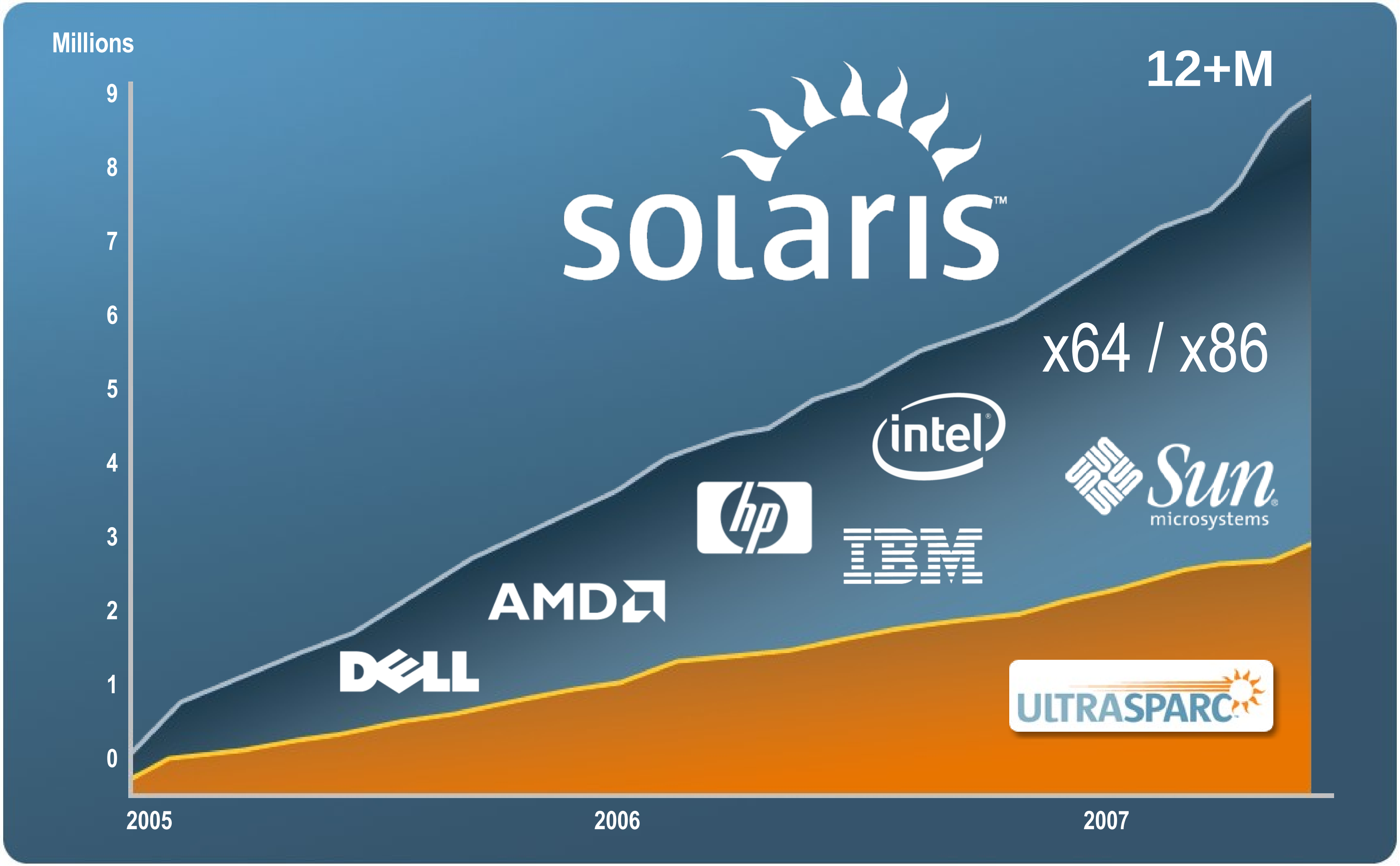
REUTERS

“MARKET OFFERING: SOLARIS RATING: **STRONG POSITIVE**”

Gartner®

Vendor Rating: Sun Microsystems
April 23, 2007

70% of Licenses on x86



Cumulative Data

OpenSolaris and Solaris

OpenSolaris

- Source code
- CDDL
- Community
  - > Supported
  - > Governance Board
  - > Sun support, too
- User Groups
- Distributions
- More rapid change
- OS development is here

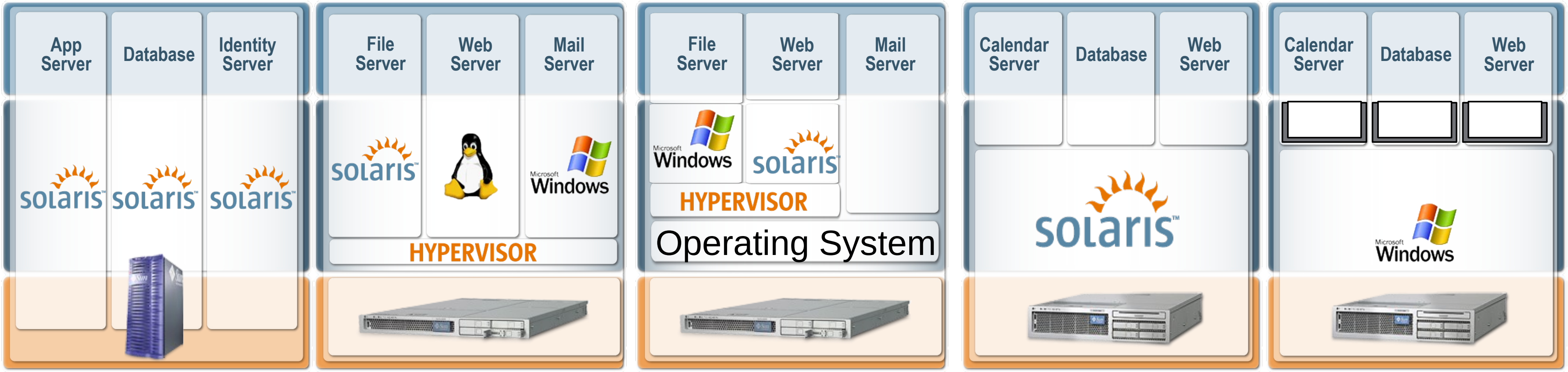
Solaris

- Binary
- Subscriptions
- Free RTU license
- Long-term Sun Support
- Sun Services
- Sun Training
- ISV application certification
- Indemnification

A brief overview of Sun virtualization technology

(with some computer architecture thrown in)

Virtualization Types



Hard Partitions

Dynamic System Domains

IBM LPAR (Mainframe)

Hypervisor: Type 1

Logical Domains

xVM Server

Xen

Vmware

z/VM

Hyper-V

HP nPar

Hypervisor: Type 2

Virtual Box

VMware Workstation

VMware Server

Microsoft Hyper-V

Parallels Workstation

OS Virtualization

Solaris Containers (Zones + SRM)

IBM WPars

Parallels Virtuozzo

Application Virtualization

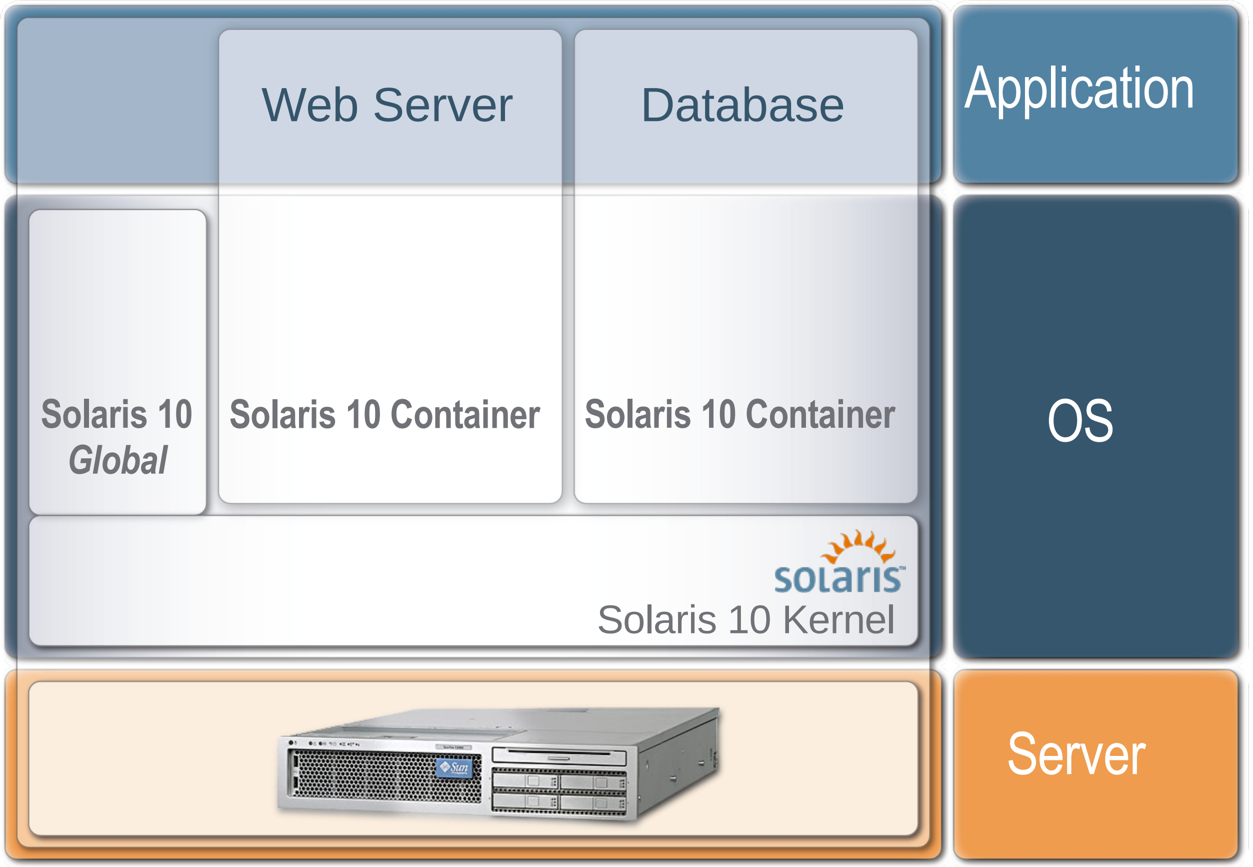
Solaris Containers for Solaris 8 & 9

Solaris Containers for Linux Applications

Microsoft SoftGrid

VMware ThinApp

Solaris Containers



Solaris Containers (“Zones”)

- OS virtualization provides virtual environments with performance, scale and observability
  - > Free feature Introduced with Solaris 10 and enhanced in each update
- Appearance of many OS instances, not many machines
  - > Isolation, integrity, security, and separate OS identities
    - – Private name, IP addresses and port ranges
    - – Private process lists and authentication (file, NIS, LDAP,...)
    - – Can boot, reboot a zone, run rc.*N* scripts
  - > Can create a new zone in minutes; takes even less via cloning
- The right way to compatibly consolidate many smaller Solaris systems
  - > Mature, widely adopted and in production at many institutions

Zone Performance

- No “virtualization penalty”
  - no emulation layer, no added latency, no cap on I/O performance, no CPU penalty, negligible memory footprint
- Scales easily to hundreds per server
  - 1,000 tested on a small server
  - Negligible overhead (CPU, RAM, disk footprint) whether idle or in use
- Integrates with resource manager
  - Granular, flexible CPU, I/O, RAM, swap space allocation:
- Intra-zone networking at memory speeds
  - Benchmarked at 18Gb/sec
- Trivial to share binaries in RAM across zones

Zone applications

- Consolidate many physical machines onto a single instance
- “Provision on demand” container for service deployment based on pre-configured system images
- Easy to clone from a pre-configured image
- Easy to migrate zone from box to box

“Branded zones” - different “OS personality”

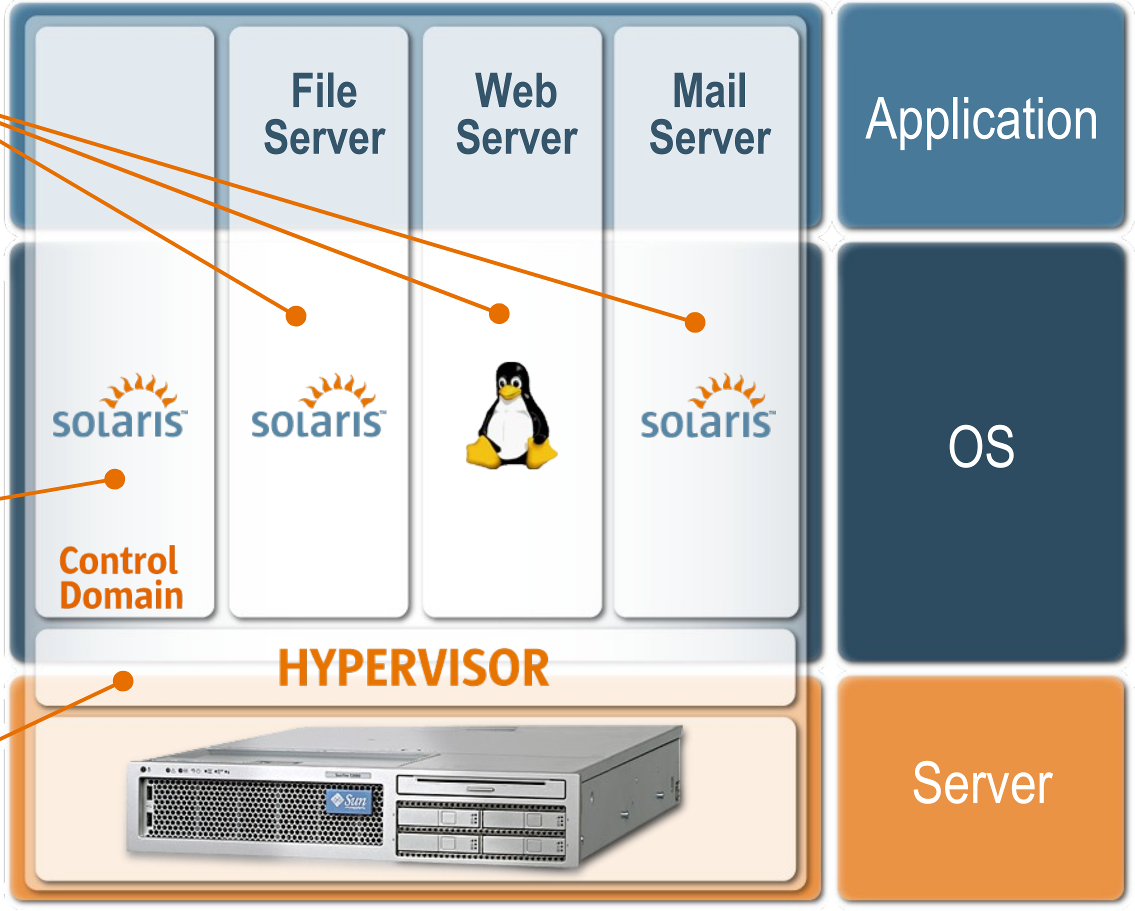
- “Linux brand” (on x86) let you run Linux applications
  - > Interposition layer remaps Linux system calls
  - > Install Linux binaries (even rpms) and libraries and run them
- “Solaris 8 Containers” (on SPARC)
  - > Most Solaris 8 apps “just work” under S10, but this provides a virtual Solaris 8 system under Solaris 10 kernel.
  - > P2V tool copies from existing system to ease the move
  - > Consolidate *many* end-of-life boxes onto the same server
  - > Resulting systems are supportable – Sun engineering tests patches for Solaris 8 in both native and container form
- Both brands let you leverage DTrace, ZFS, and other features of Solaris while in encapsulated state

Logical Domains

Solaris or Linux guest domains

Solaris Control Domain

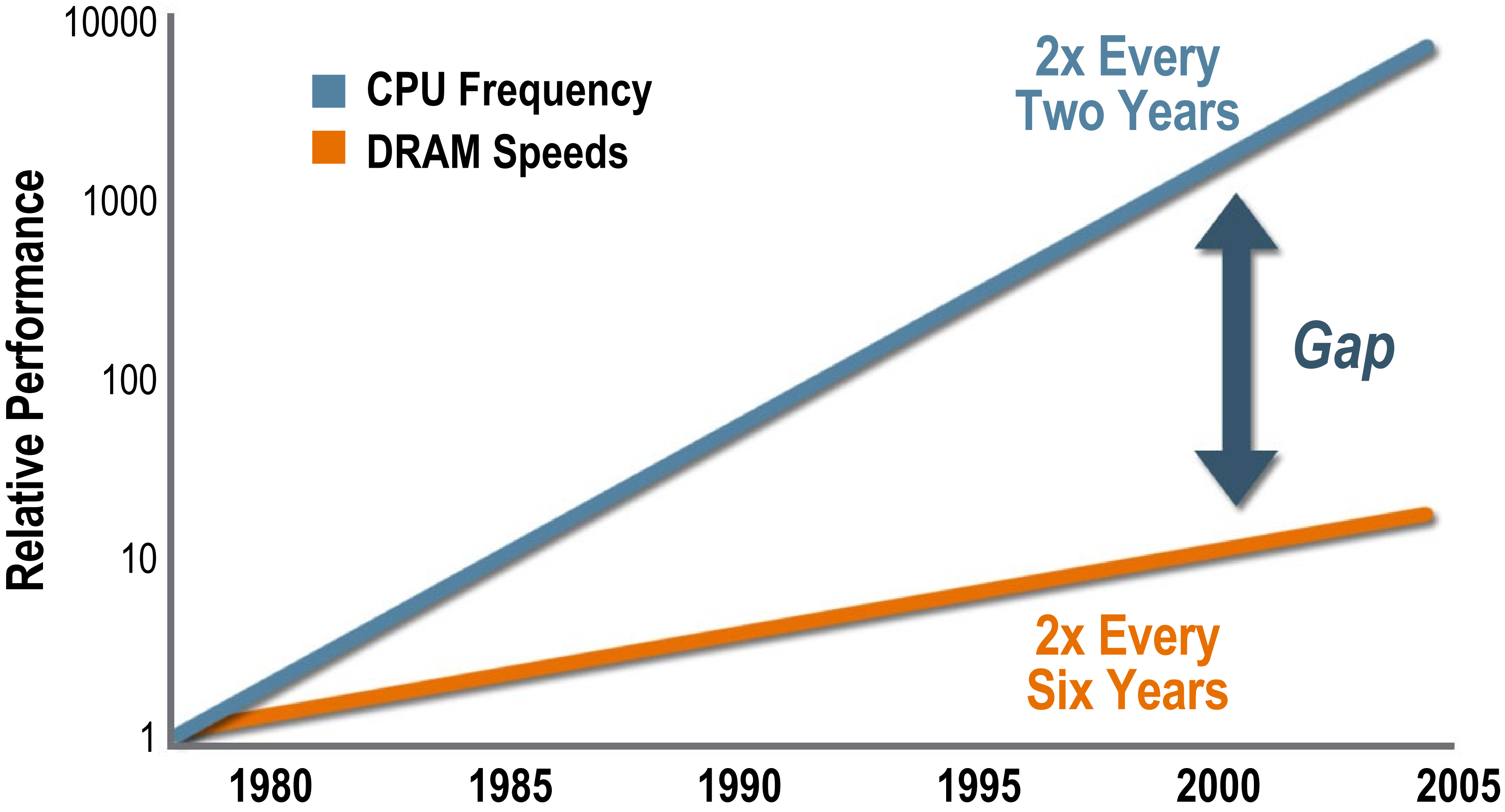
Ultra lightweight Hypervisor in the firmware



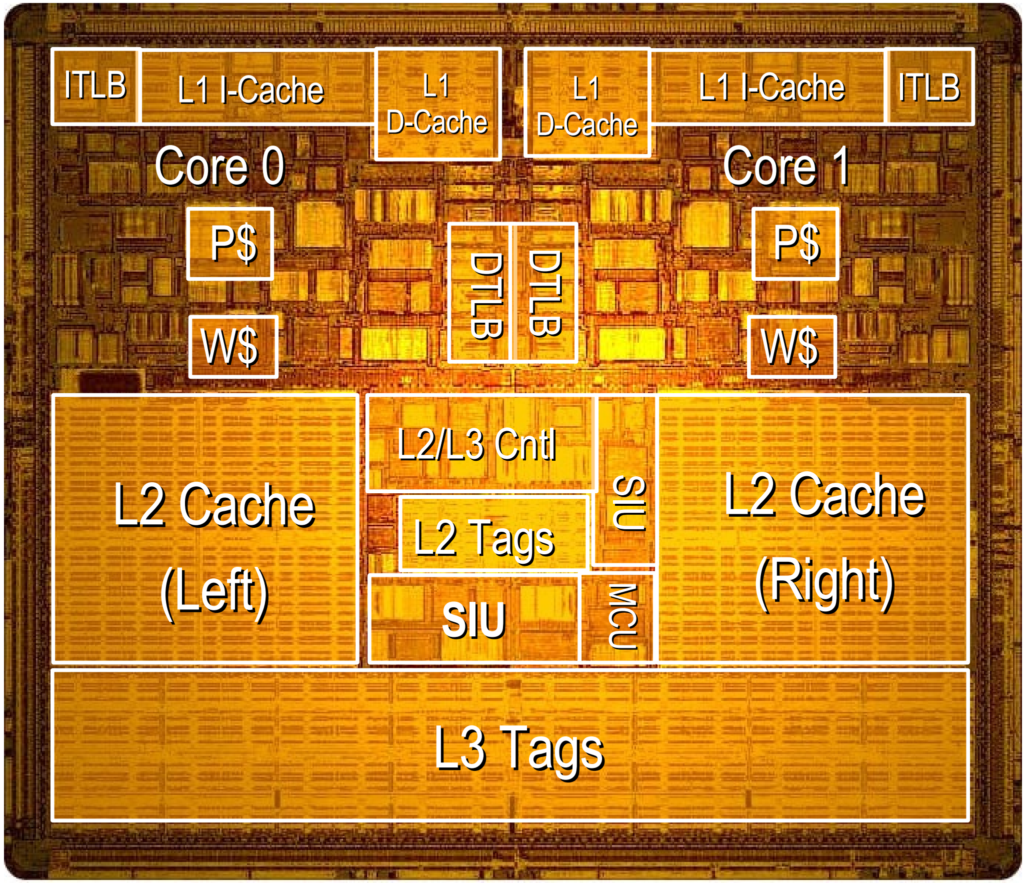
A little chip talk first...

- Logical domains do virtualization differently (on purpose, of course)
- A little hardware background is in order, so let's talk about contemporary computer architecture issues

The Memory Bottleneck



How We Mask Memory Latency Today

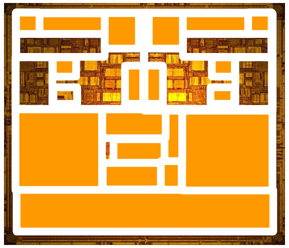


Great Big Caches

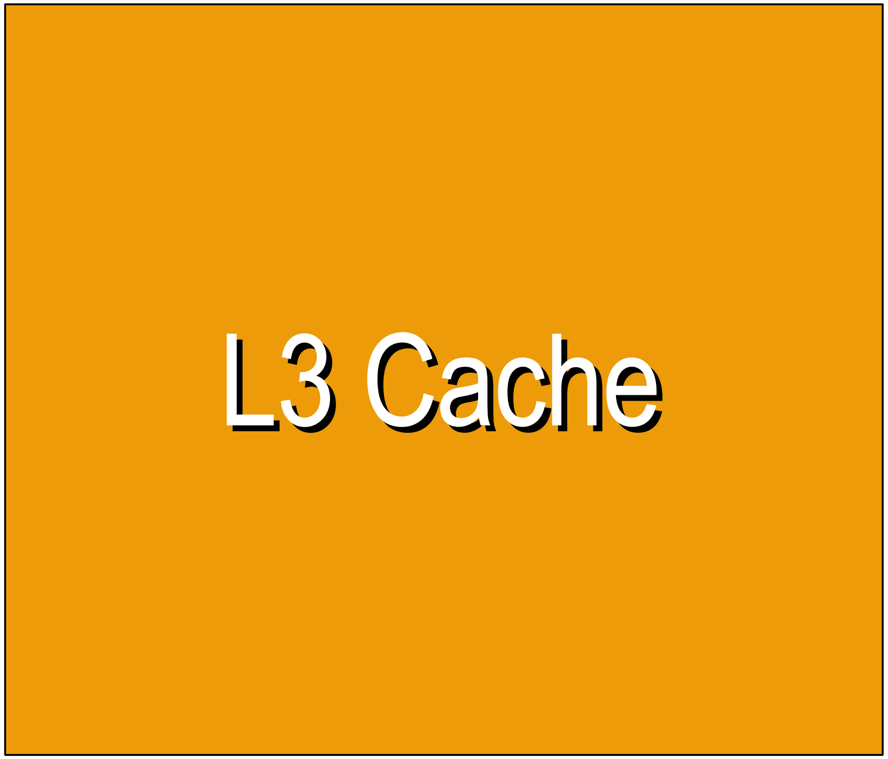
- of many different kinds
- that only mask memory latency
- and execute no code
- accessed one cache line at a time
- but require power and cooling to all lines all the time

Cache Logic Accounts for About 75% of the Chip Area

This Is Getting Ridiculous!



295 Million Transistors



L3 Cache

~ 2 Billion Transistors

- Can you spot the transistors that are actually executing code?
- Unfortunately, big caches are essential for ILP processors (they have nothing else to mask memory latency)

Can We Re-Think Processor Design to Do Something Smarter with Our Transistors?

Clock speed and power consumption

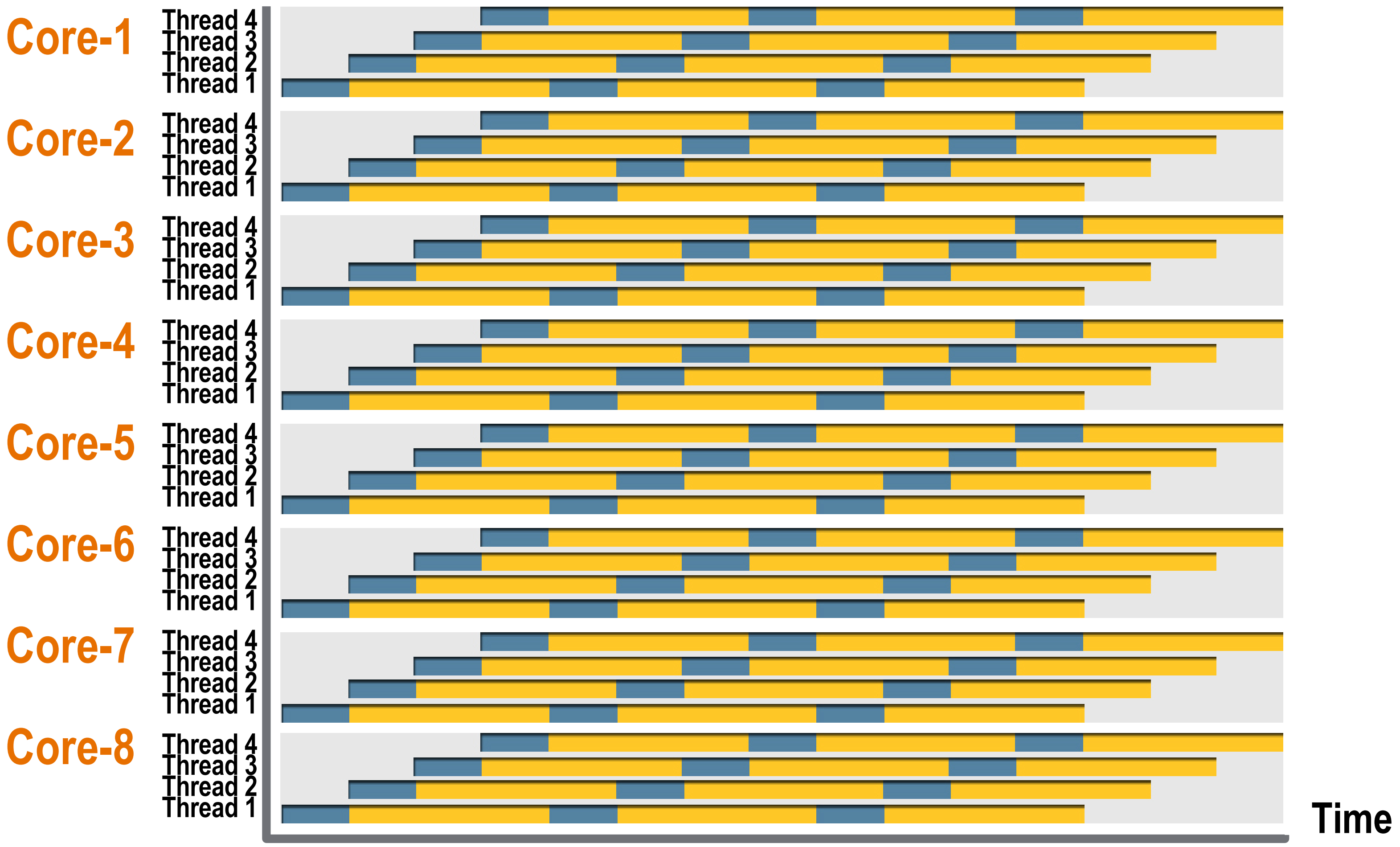
- Distributing the system clock consumes ~25% of the power in a typical processor core
- Extremely high clock rates are no longer crucial for performance – the world is going multi-core

Chip Multi-Threading (CMT)

Significantly Higher Throughput from a Team of Multi-threaded Processor Cores

🟨 Memory Latency

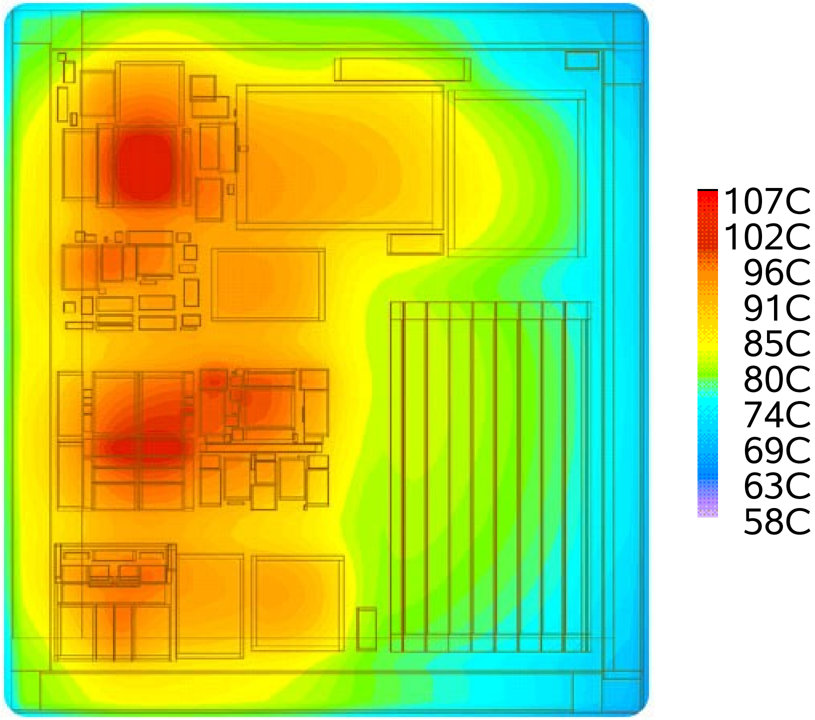
🟦 Compute

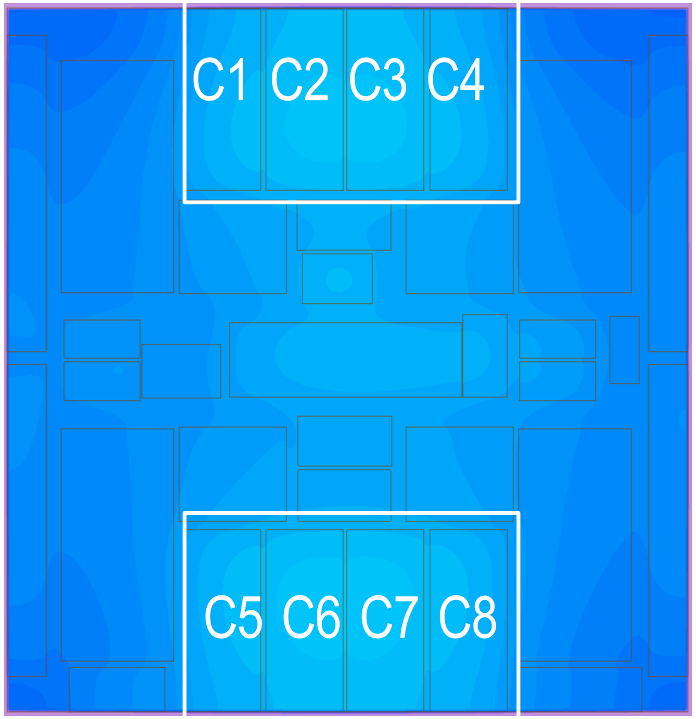


CMT Power Advantage

“Cool Threads” Dramatically Reduce Power Consumption

Uses a Fraction of the Power/Thread





General Purpose Processor *(Size Not to Scale)* CMT Processor

Results with Chip Multithreading

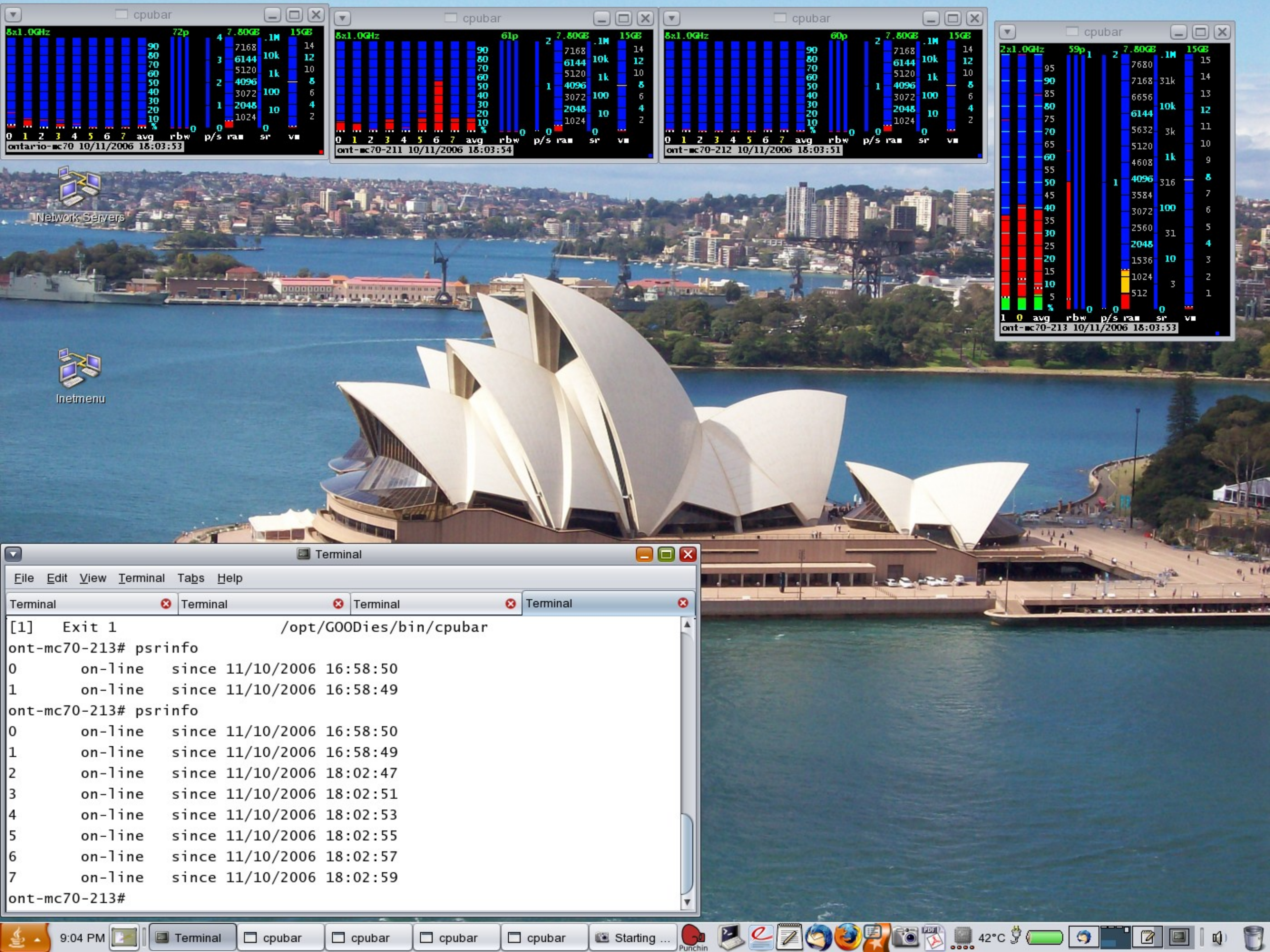
- Current Sun products have up to 256 CPU threads
  - > No single thread runs very fast, with clock <2GHz (we provide single thread speed in our enterprise line), but
  - > You get many of them, and they run in parallel
  - > Switch thread within a single clock on cache miss
- Very low power load – due to the low frequency and integrated NICs, crypto and other on-chip features
- What do you do with 32, 64, 128 or 256 threads?
  - > Run parallel apps: Java, web, messaging, even DBMS
  - > Run multiple instances of serial apps
  - > **Run virtual machines**

Logical Domains

- Free virtual machine capability for SPARC CMT
- Each domain is an entirely independent machine with its own
  - CPUs, RAM, hardware crypto accelerators
  - Virtual disks
  - console
  - network interfaces, MAC and IP addresses
- Each domain has its own
  - OS kernel, patches, tuning parameters
  - user accounts, administrators
  - Easily cloned from “golden images”

Virtual CPUs in logical domains

- Chip Multi Threading servers have up to 128 virtual CPUs (aka threads) in 1RU or 2RU; 256 in 4RU
- A domain can have any number of threads, 1 to “all”
  - Each belongs to the domain, so no overhead enabling or disabling interrupts, changing memory mapping, etc
- Getting a lot of adoption for consolidation
  - Can, and should, use zones inside a domain
- Can be dynamically allocated with the domain running. Adding or removing a vcpu to or from a running domain takes effect immediately


cpubar
8x1.0GHz
72p
7.80GB
15GB
ontario-mc70 10/11/2006 18:03:53
cpubar
8x1.0GHz
61p
ont-mc70-211 10/11/2006 18:03:54
cpubar
8x1.0GHz
60p
ont-mc70-212 10/11/2006 18:03:51
cpubar
2x1.0GHz
59p
ont-mc70-213 10/11/2006 18:03:53
0 1 2 3 4 5 6 7 avg rbw p/s ram sr vm
Network Servers
Inetmenu
Terminal
File Edit View Terminal Tabs Help
Terminal
Terminal
Terminal
Terminal
9:04 PM
Terminal
cpubar
cpubar
cpubar
cpubar
Starting ...
42°C

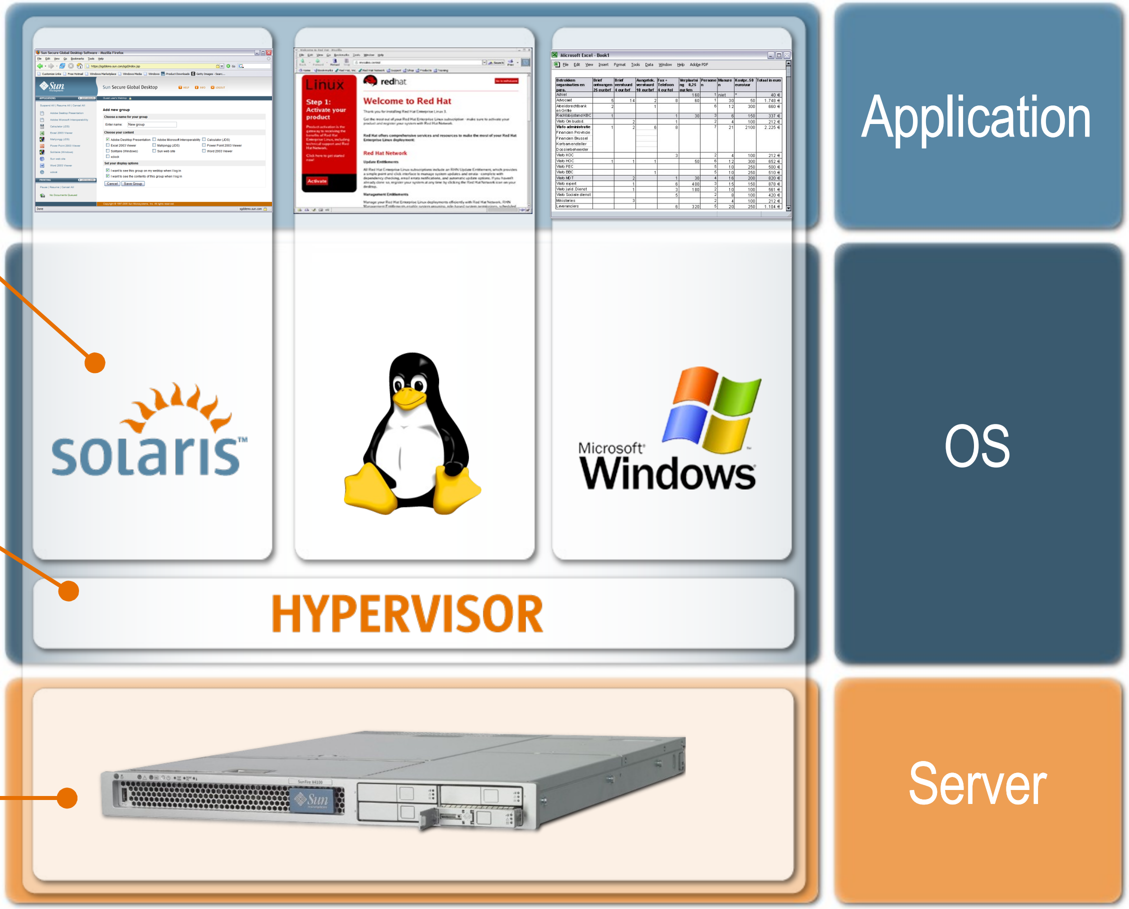
```
[1]    Exit 1                    /opt/GOODies/bin/cpubar
ont-mc70-213# psrinfo
0       on-line   since 11/10/2006 16:58:50
1       on-line   since 11/10/2006 16:58:49
ont-mc70-213# psrinfo
0       on-line   since 11/10/2006 16:58:50
1       on-line   since 11/10/2006 16:58:49
2       on-line   since 11/10/2006 18:02:47
3       on-line   since 11/10/2006 18:02:51
4       on-line   since 11/10/2006 18:02:53
5       on-line   since 11/10/2006 18:02:55
6       on-line   since 11/10/2006 18:02:57
7       on-line   since 11/10/2006 18:02:59
ont-mc70-213#
```

Virtual Machines

Allows different OS versions and types

Extra overhead for the Hypervisor

Available on many platforms



Sun xVM Server Family

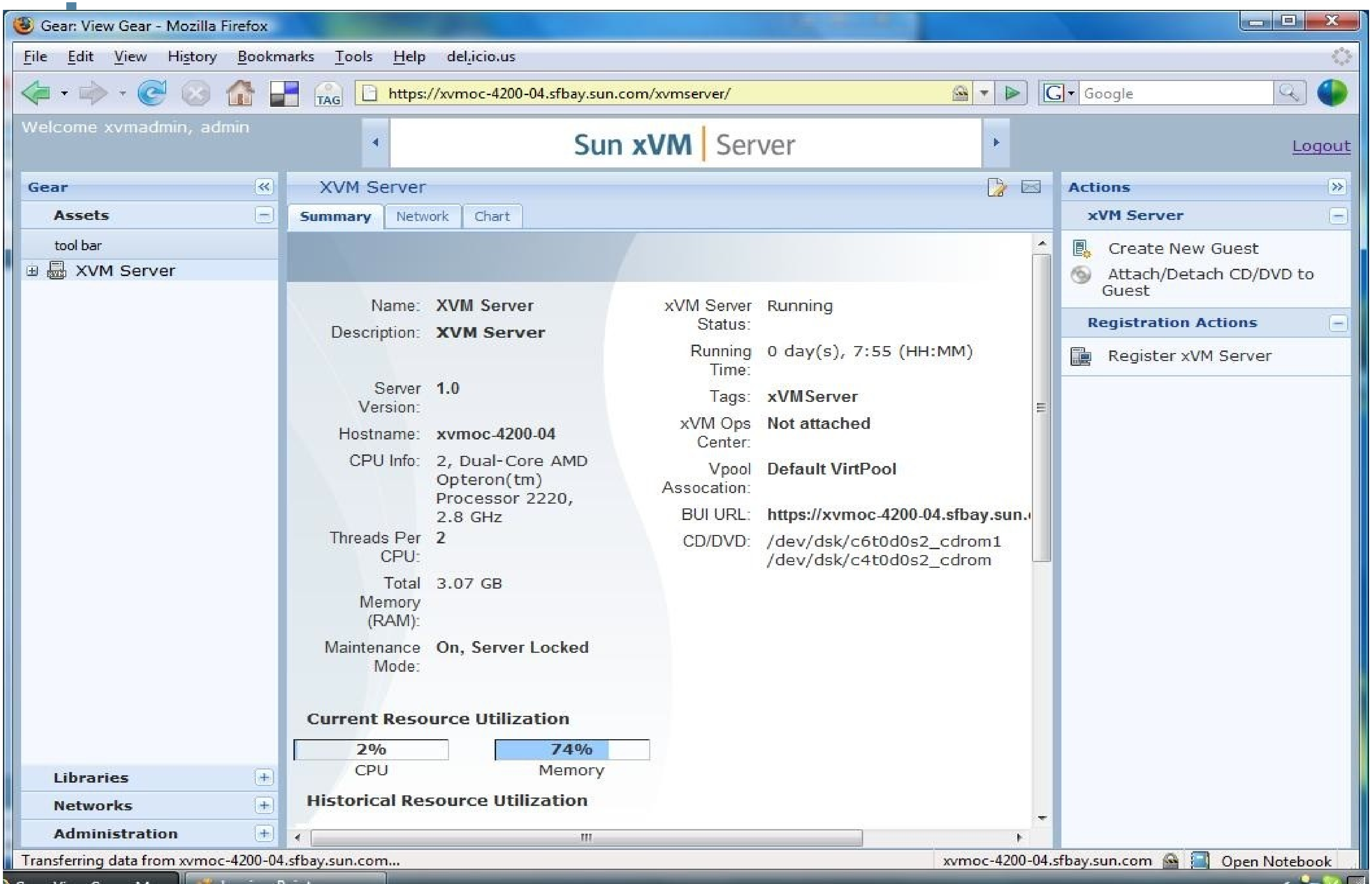
SUN xVM SERVER

- For Enterprises to deploy in Datacenters
- Support on x86 and SPARC
- Advanced Enterprise features include Live Migration, Predictive-Self Healing, Advanced I/O and Security
- Support for Sun xVM Ops Center

SUN xVM VIRTUALBOX

- For Developers to use on Desktops and Laptops
- Type 2 Hypervisor for x86 only
- Cross platform support for Windows, Linux, Mac OS and Solaris
- No live migration; Supports USB on remote RDP sessions
- Free 17MB download

xVM Hypervisor in Action: hardware



Name: XVM Server

Description: XVM Server

Server Version: 1.0

Hostname: xvmoc-4200-04

CPU Info: 2, Dual-Core AMD Opteron(tm) Processor 2220, 2.8 GHz

Threads Per CPU: 2

Total Memory (RAM): 3.07 GB

Maintenance Mode: On, Server Locked

xVM Server Status: Running

Running Time: 0 day(s), 7:55 (HH:MM)

Tags: xVMServer

xVM Ops Center: Not attached

Vpool Assocation: Default VirtPool

BUI URL: https://xvmoc-4200-04.sfbay.sun.

CD/DVD: /dev/dsk/c6t0d0s2_cdrom1 /dev/dsk/c4t0d0s2_cdrom

Current Resource Utilization

2% CPU

74% Memory

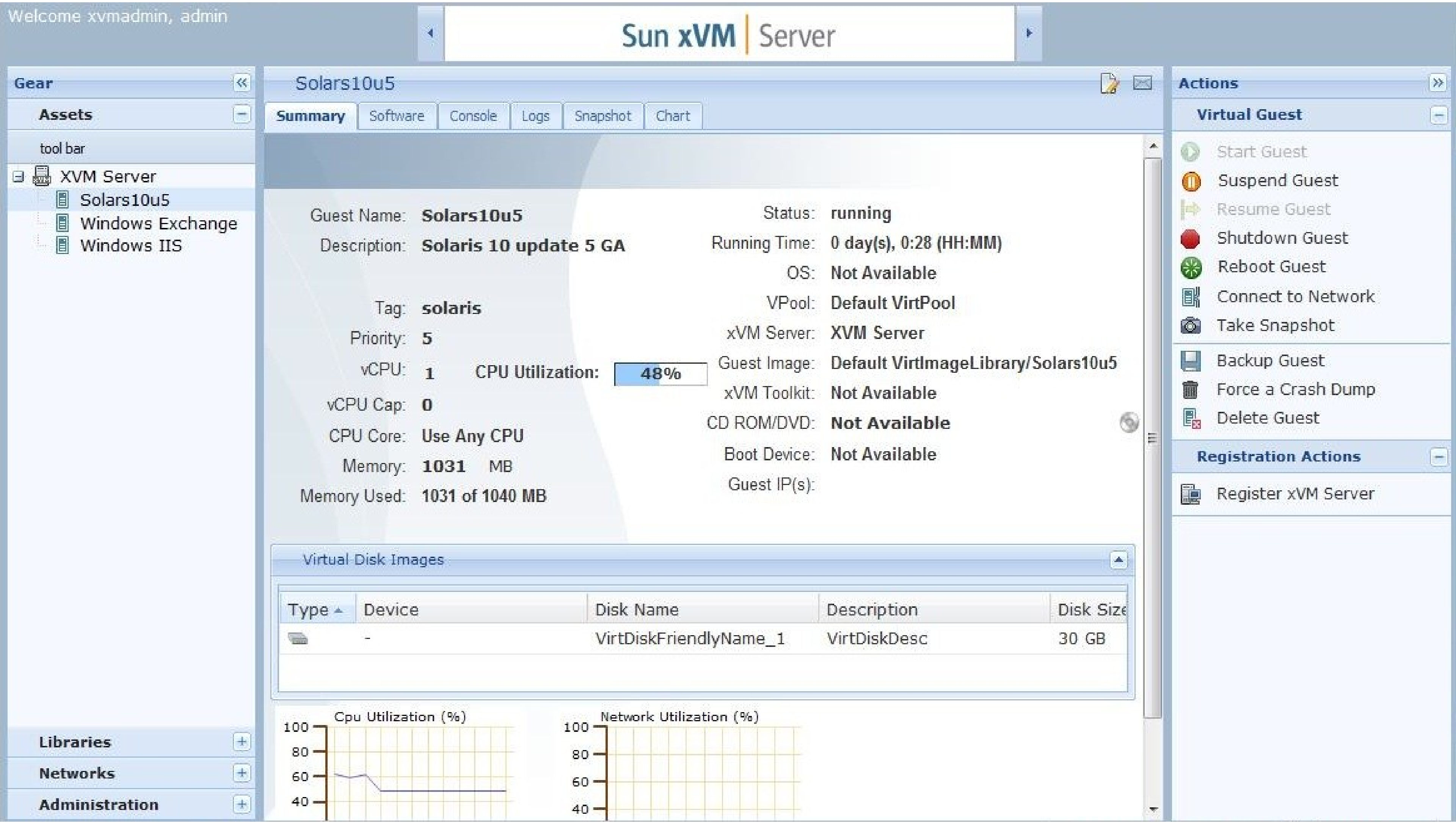
Historical Resource Utilization

Transferring data from xvmoc-4200-04.sfbay.sun.com...

xvmoc-4200-04.sfbay.sun.com

Open Notebook

xVM Hypervisor in Action: guest view



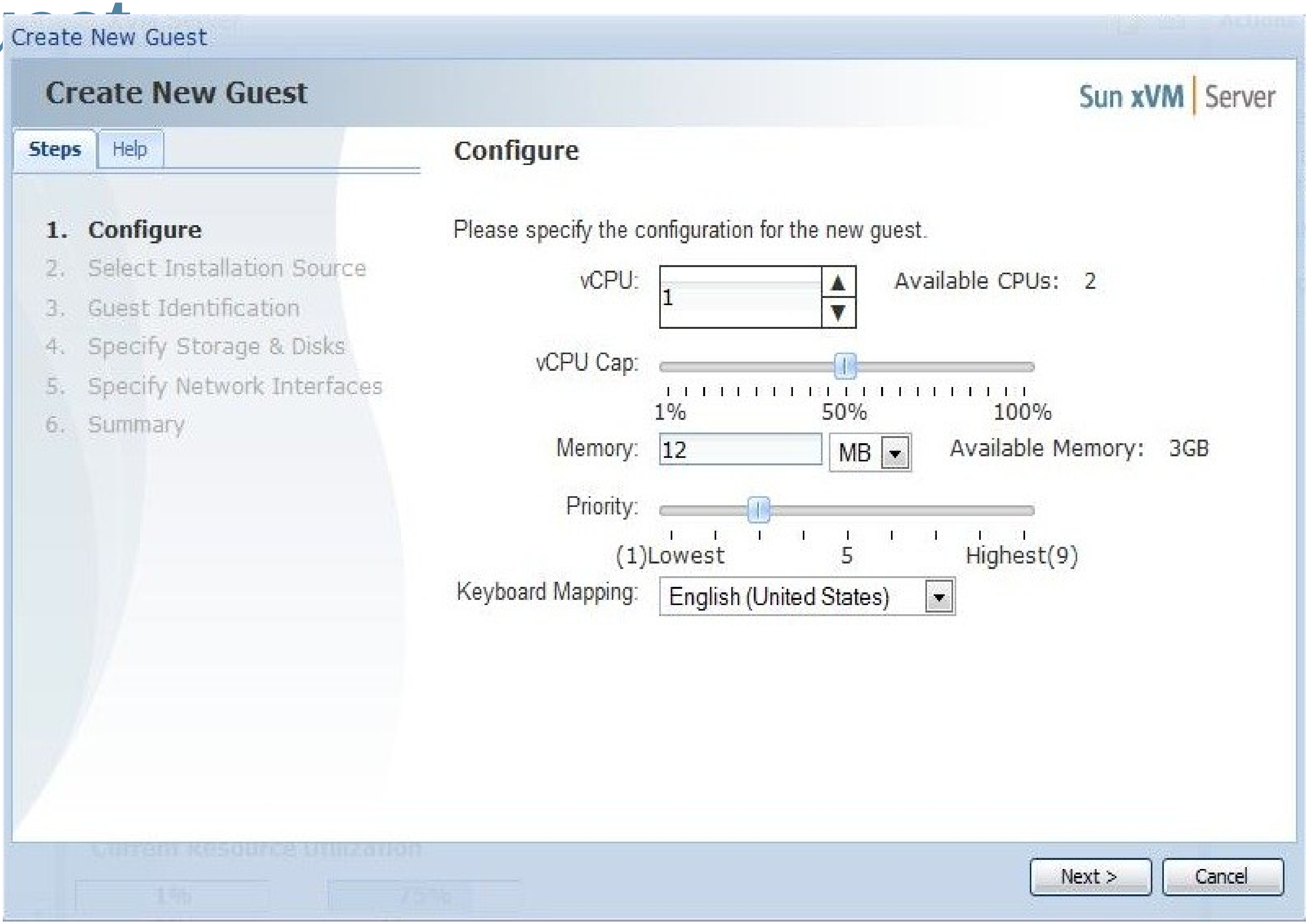
Guest Name: Solars10u5
Description: Solaris 10 update 5 GA
Tag: solaris
Priority: 5
vCPU: 1 CPU Utilization: 48%
vCPU Cap: 0
CPU Core: Use Any CPU
Memory: 1031 MB
Memory Used: 1031 of 1040 MB

Status: running
Running Time: 0 day(s), 0:28 (HH:MM)
OS: Not Available
VPool: Default VirtPool
xVM Server: XVM Server
Guest Image: Default VirtImageLibrary/Solars10u5
xVM Toolkit: Not Available
CD ROM/DVD: Not Available
Boot Device: Not Available
Guest IP(s):

Virtual Disk Images

Type ▲	Device	Disk Name	Description	Disk Size
	-	VirtDiskFriendlyName_1	VirtDiskDesc	30 GB

xVM Hypervisor in Action: create guest



Create New Guest

Create New Guest

Sun xVM | Server

1. **Configure**
2. Select Installation Source
3. Guest Identification
4. Specify Storage & Disks
5. Specify Network Interfaces
6. Summary

Configure

Please specify the configuration for the new guest.

vCPU: 1

Available CPUs: 2

vCPU Cap: 1% 50% 100%

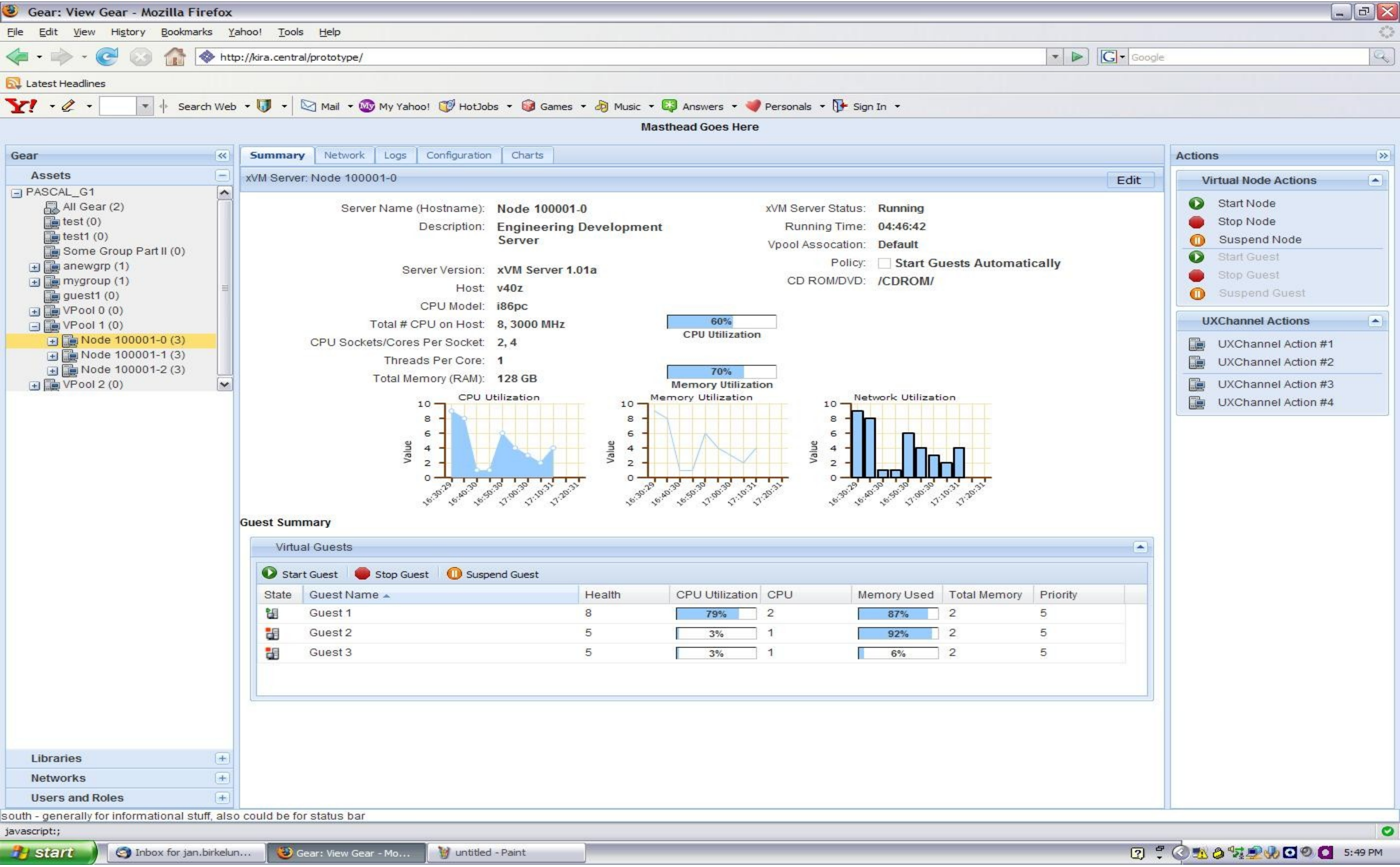
Memory: 12 MB

Available Memory: 3GB

Priority: (1)Lowest 5 Highest(9)

Keyboard Mapping: English (United States)

Management console



xVM Server: Node 100001-0

Server Name (Hostname): Node 100001-0
Description: Engineering Development Server
Server Version: xVM Server 1.01a
Host: v40z
CPU Model: i86pc
Total # CPU on Host: 8, 3000 MHz
CPU Sockets/Cores Per Socket: 2, 4
Threads Per Core: 1
Total Memory (RAM): 128 GB

xVM Server Status: Running
Running Time: 04:46:42
Vpool Assocation: Default
Policy: □ Start Guests Automatically
CD ROM/DVD: /CDROM/

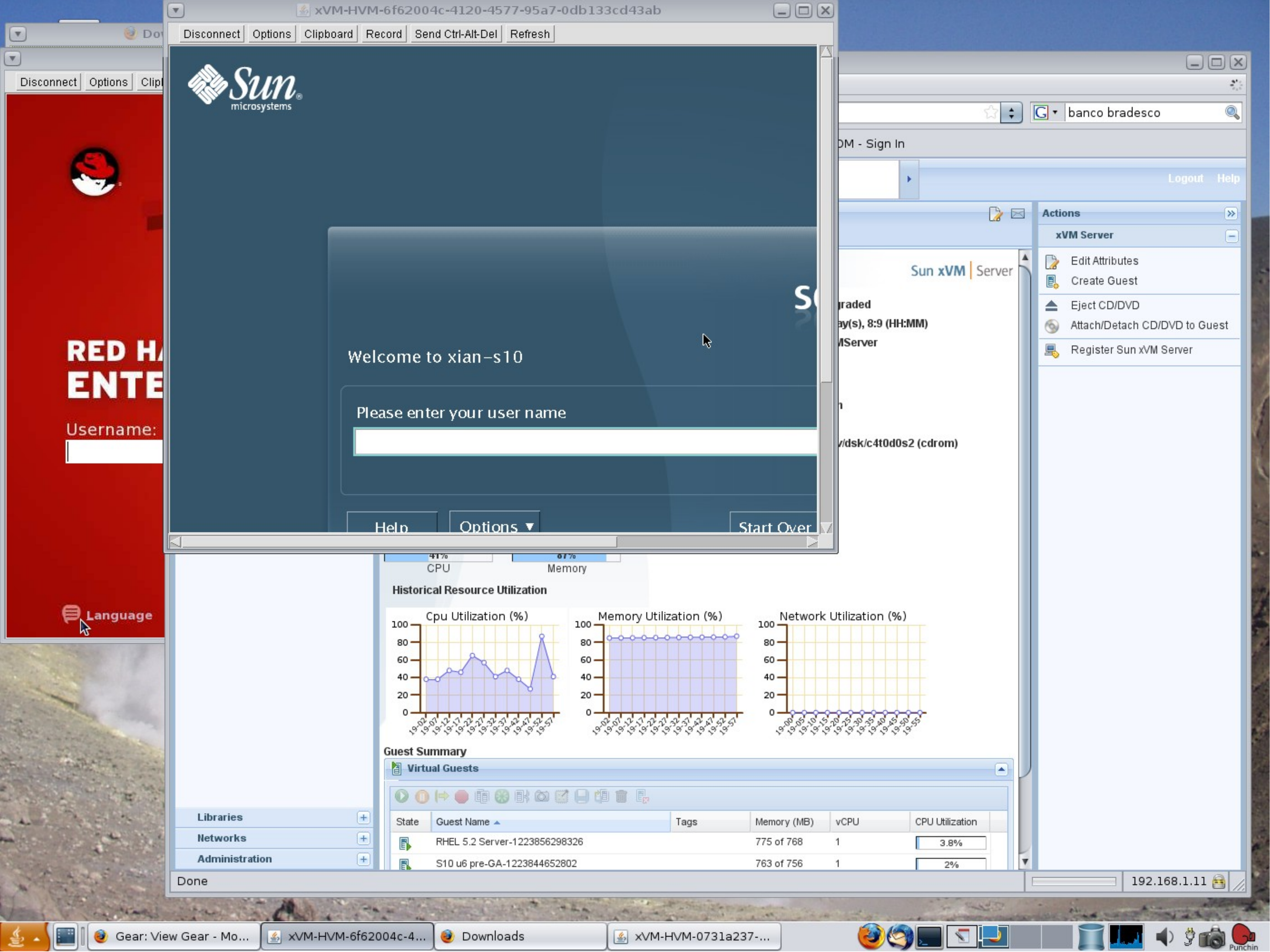
Guest Summary

Virtual Guests

State	Guest Name	Health	CPU Utilization	CPU	Memory Used	Total Memory	Priority
	Guest 1	8	79%	2	87%	2	5
	Guest 2	5	3%	1	92%	2	5
	Guest 3	5	3%	1	6%	2	5

south - generally for informational stuff, also could be for status bar

javascript:;



Welcome to xian-s10

Please enter your user name

graded

ay(s), 8:9 (HH:MM)

MServer

/dsk/c4t0d0s2 (cdrom)

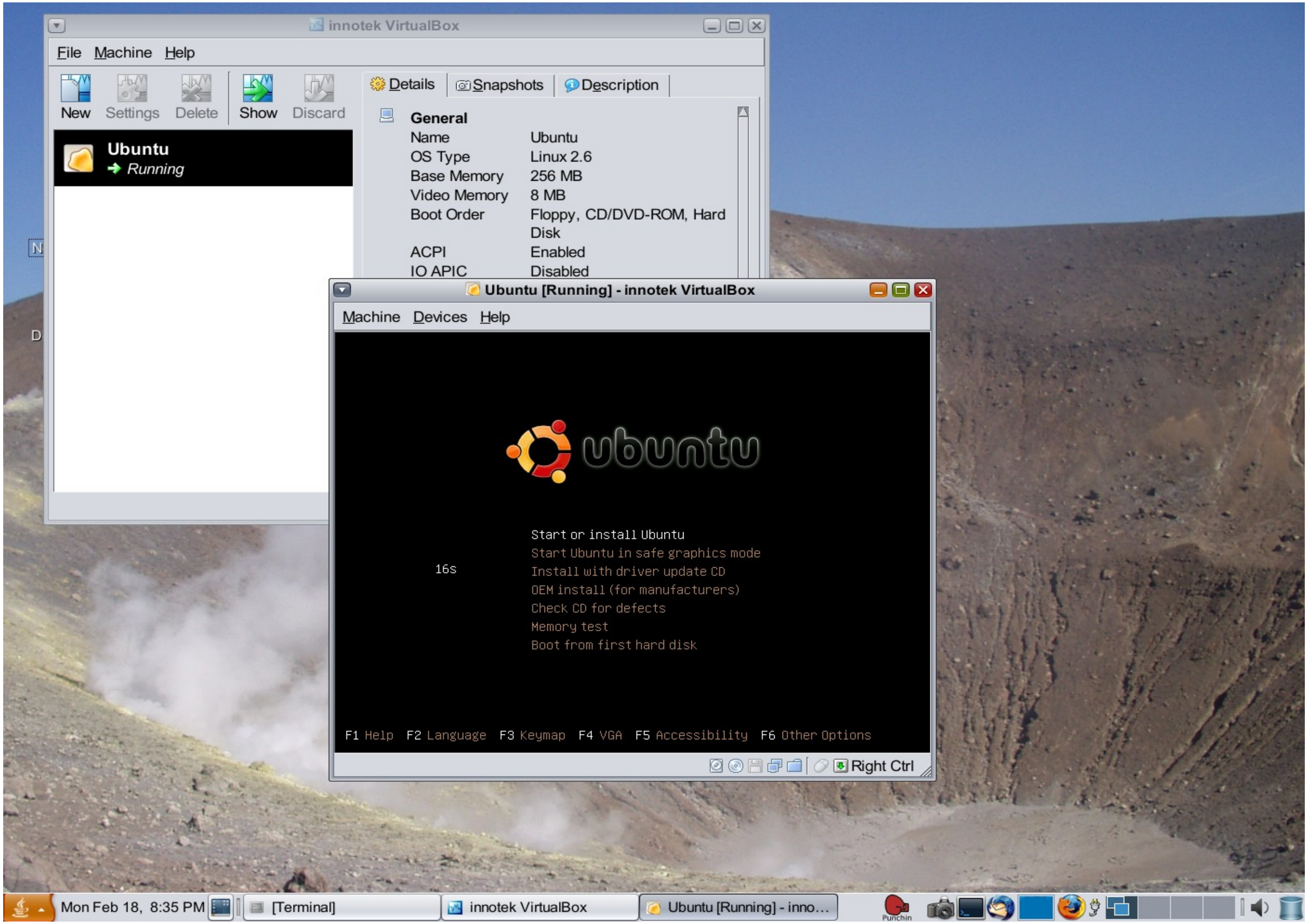
Historical Resource Utilization

Guest Summary

Virtual Guests

State	Guest Name ▲	Tags	Memory (MB)	vCPU	CPU Utilization
	RHEL 5.2 Server-1223856298326		775 of 768	1	3.8%
	S10 u6 pre-GA-1223844652802		763 of 756	1	2%

Installing a guest under VirtualBox



General

Name	Ubuntu
OS Type	Linux 2.6
Base Memory	256 MB
Video Memory	8 MB
Boot Order	Floppy, CD/DVD-ROM, Hard Disk
ACPI	Enabled
IO APIC	Disabled

```
                  Start or install Ubuntu
                  Start Ubuntu in safe graphics mode
     16s          Install with driver update CD
                  OEM install (for manufacturers)
                  Check CD for defects
                  Memory test
                  Boot from first hard disk

F1 Help  F2 Language  F3 Keymap  F4 VGA  F5 Accessibility  F6 Other Options
```

What is OpenSolaris on System z?

- A port of OpenSolaris to System z and z/VM by Sine Nomine Associates
  - Based on OpenSolaris source distro
- Solaris already runs on SPARC and Intel+AMD
  - A port effort is also under way for POWER
- Note these URLs:
  - http://www.opensolaris.org/os/project/systemz/
  - http://distribution.sinenomine.net/opensolaris
- Join SOL-390 at vm.marist.edu

In the beginning...

- September 2005:
  - > “Wanted to get your opinion of an idea. In our spare time (hah), we've been toying with the idea of doing a 390 port of OpenSolaris.” db
- In design discussions with Neale, I encouraged:
  - > Make it 64-bit only. 31-bit is so last-century (there's no legacy Solaris 31-bit code on z to be compatible with)
  - > Target z/VM, not bare metal (saves a great amount of coding and testing effort that probably would be wasted)
    - – Use DIAGNOSE for I/O, let CP do error handling, etc.
  - > [I'm sure Neale didn't actually *need* any advice! :-)]

My participation

- Introduced Neale to Greg Papadopoulos, Sun's CTO and EVP of R&D in May 2006
  - > I went to Greg's office, Neale was on the phone
  - > Greg sees value in Solaris on other platforms. General Sun policy is to encourage OpenSolaris everywhere
- This led to my being able to help obtain the loaner Sun workstation Neale used for the code port
- Made Sun introductions for David and Neale
- I installed and tested on z9 inside Sun STK, sending comments and bug reports to SNA

Install details

- I started testing OpenSolaris on z in February 2008
- Guest with 512MB and 2 virtual CPUs
- Userid that runs installer needs SAVESEG privilege
- Install from AWSTAPE file and unload VMARC files, or direct from VMARC files containing DDR images
  - > Use DDR2CMSX to DDR from a CMS file to disk.
- 200 disk for Solaris system volume
- 191 disk recomped to have IPL cylinders. You “IPL 191” just like RSCS in the Good Old Days

A side note...

- It's nice to have a VM userid again
  - Actually, I've never stopped having a VM userid, but class G userids are boring, and I had no particular reason to logon to CMS
- Once the VM guys decided I wasn't a VM noob, they gave me more privilege classes :-)
- Fun learning new commands such as the 64-bit display commands
- Yes, I still remember how to use VM...

A side note, part 2

- Embarrassing to occasionally issue the wrong form of pipe:
  - `q ALLOC|grep SPOOL` **does not work! :-)**
  - `pipe cp q alloc|locate /SPOOL/|console` **does work**
- I've even typed “`cat profile exec`” which is *really* embarrassing.
  - OTOH, I still “`cat any.file|take last 10`” on Solaris or Linux once in a while. What a mess.
- Lesson: Hands have their own habits

First impressions

- This is impressive engineering. One (mostly) or several people porting an OS is a big accomplishment
- If you know how to login in line-mode and use Solaris, AIX or Linux commands, you'll feel pretty much at home
- The software available consists of
  - Standard OS-provided commands
  - C compiler (gcc)
  - A web server, lighthttpd

Testing history

- At beginning of 2008: could boot, no network
  - > Working from a virtual 3215 is too painful for real use
- June 2008: Multi-user and network.
  - > Requires minimum of a z9 at z/VM 5.3 + VM64466
  - > Some delay getting the APAR
- August 2008: New “phase”
  - > I started doing some serious testing
- October 2008: public binary drop
  - > http://distribution.sinenomine.net/opensolaris

Networking

- z/VM 5.3 + VM64466 provided network access
- OSA support only, requiring new DIAG in APAR
- 'CP DEFINE NIC 340 TYPE QDIO'
- 'CP COUPLE 340 SYSTEM' server_vmid
- Now I could ssh in and have reasonable CLI access

Disk

- All disks via minidisks, using CMS FORMAT and RESERVE
- Disk I/O via DIAGNOSE
- Each disk appears to Solaris as
  - > “/dev/disk/c”||x2d(device_address)|”s3”. For example, disk at address 0200 is /dev/disk/c0d512s3
  - > Device is a link to /devices/ccw/: For example: /devices/ccw/dasd@0x0300:dasd

Let's have a look (guest console)

```
00: q v all
00: STORAGE = 512M
00: XSTORE = none
00: CPU 00  ID  FF08D09C20948000 (BASE) STOPPED CP   CPUAFF ON
00: CPU 01  ID  FF08D09C20948000 STOPPED CP   CPUAFF ON
00: No AP Crypto Queues are available
00: CONS 0009 ON LDEV L0029   TERM STOP  HOST TCPIP    FROM 129.150.49.121
00:      0009 CL T NOCONT NOHOLD COPY 001    READY FORM STANDARD
00:      0009 TO ZIP00SOL PRT DIST J_SAVIT   FLASHC 000 DEST OFF
00:      0009 FLASH       CHAR       MDFY       0 FCB       LPP OFF
00:      0009 3215   NOEOF CLOSED    NOKEEP NOMSG NONAME
00:      0009 SUBCHANNEL = 0005
...
00: DASD 0190 3390 V53ZAR R/O    107 CYL   0213 RELN ON DASD  6E0B SUBCHANNEL = 0009
00: DASD 0191 3390 VUSZA1 R/W     50 CYL   3080 RELN ON DASD  6E09 SUBCHANNEL = 0000
00: DASD 019D 3390 ZVM530 R/O    115 CYL   1548 RELN ON DASD  44D6 SUBCHANNEL = 000A
00: DASD 019E 3390 V53ZAR R/O    250 CYL   0935 RELN ON DASD  6E0B SUBCHANNEL = 000B
00: DASD 0200 3390 VUSZA1 R/W   3338 CYL   3160 RELN ON DASD  6E09 SUBCHANNEL = 0004
00: DASD 0201 3390 VUSZA2 R/O   3338 CYL   3339 RELN ON DASD  6E0A SUBCHANNEL = 0002
00: DASD 0300 9336 (VDSK) R/W 600064 BLK   0000 RELN ON DASD  VDSK SUBCHANNEL = 0003
00: DASD 0319 3390 VMPP02 R/O     75 CYL   0898 RELN ON DASD  4433 SUBCHANNEL = 000C
00: DASD F200 3390 VUSZA2 R/W   3338 CYL   0001 RELN ON DASD  6E0A SUBCHANNEL =
                                                            MORE...   ZIPAVM
```

Let's have a look (guest console)

```
00: ipl 191 cl
00: Boot commenced for kernel built on Jul 11 2008 10:00:45
00: initialize scratch memory
00: Installed physical memory @ 4400000:
00: (0x00, 0x020000000)
00: Booter occupied memory (including modules) @ 4400060:
00: (0x0100000, 0x0167000)(0x04400000, 0x0800000)
00: Ramdisk memory @ 4400080:
00: (0x02000000, 0x02400000)
00: Available physical memory @ 4400100:
00: (0x0267000, 0x04199000)(0x04c00000, 0x01b400000)
00: Free physical memory @ 44000e0:
00: (0x0267000, 0x01d99000)(0x04c00000, 0x01b400000)
00: Available virtual memory @ 44000c0:
00: (0x00, 0x0100000)(0x0267000, 0x01d99000)(0x04c00000, 0xffffffffb3fffff)
00: DAT Enabled using RTO 4c00000
00: Creating mappings for KPM
00:         Mapping ffffffff80000000 to 0 for 512MB
00: Relocating the KRTLD/UNIX executable
...
... [many other exciting messages]
...
console login:
```

Let's have a look

```
sirius ~ $ uname -a
SunOS sirius 5.11 home/neale/OpenSolaris/ibm/onnv-gate s390x s390 s390x
sirius ~ $ df -h
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c0d512s3      2.2G   2.1G    94M    96%    /
/devices                 0K     0K     0K     0%    /devices
/dev                     0K     0K     0K     0%    /dev
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   219M   536K   218M     0%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
fd                       0K     0K     0K     0%    /dev/fd
swap                   218M     0K   218M     0%    /tmp
swap                   218M    28K   218M     0%    /var/run
sirius ~ $ mpstat 30
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0    0   0    0     0    0    0    0    0    0    0     0    0  99   0   0
  1    0   0    0     0    0    0    0    0    0    0     0    0  99   0   0
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0   27  11    0   982  960  116   34   19    4    0   746   68   0   0  31
  1   13  36    0   490  407  207    9   15    3    0   667   22  12   0  64
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0   50  11    0  1005  964  207    4   19    8    0   976    1   0   0  98
  1   29  35    0   486  401  170   29   18    8    0  1545   80  15   0   4
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0   12   3    0   927  891  153   16   15   11    0   469   35   0   0  64
  1   25  23    0   465  407  106   20   14    9    0   811   55  12   0  32
```

Let's have a look

```
sirius ~ $ isalist
s390x z9 s390
sirius ~ $ prtconf
System Configuration:  IBM Corporation  s390x
Memory size: 512 Megabytes
System Peripherals (Software Nodes):
s390x
    ramdisk, instance #0
    pseudo, instance #0
    options, instance #0
    ccw, instance #0
        dasd, instance #1 (driver not attached)
        dasd, instance #2
        dasd (driver not attached)
        dasd, instance #4
        dasd, instance #5
        cnsl, instance #0
        dasd (driver not attached)
        dasd (driver not attached)
        dasd (driver not attached)
        dasd (driver not attached)
        dasd, instance #10
        osa, instance #0
        osa, instance #1
        osa, instance #2
        diag250, instance #0 (driver not attached)
    cpus (driver not attached)
```

Adding some disk space – part 1

```
* from CMS before booting Solaris:
#cp define t3390 1aa cyl 100
DASD 01AA DEFINED
Ready;
 format 1aa t
DMSFOR603R FORMAT will erase all files on disk T(1AA). Do you wish to continue?
Enter 1 (YES) or 0 (NO).
 1
DMSFOR605R Enter disk label:
 dsk1aa
DMSFOR733I Formatting disk T
DMSFOR732I 100 cylinders formatted on T(1AA)
Ready;
 reserve solaris data t
DMSRSV603R RESERVE will erase all files on disk T(1AA). Do you wish to continue?
 Enter 1 (YES) or 0 (NO).
 1
DMSRSV733I Reserving disk T
Ready;
 rel t
Ready;
```

Adding some disk space – part 2

```
sirius / # newfs /dev/dsk/c0d768s3
newfs: construct a new file system /dev/rdsk/c0d768s3: (y/n)? y
/dev/rdsk/c0d768s3:     598800 sectors in 998 cylinders of 1 tracks, 600 sectors
        292.3MB in 63 cyl groups (16 c/g, 4.68MB/g, 2240 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 9632, 19232, 28832, 38432, 48032, 57632, 67232, 76832, 86432,
 508832, 518432, 528032, 537632, 547232, 556832, 566432, 576032, 585632, 595232
sirius / # mkdir /mnt/disk300
sirius / # mount /dev/dsk/c0d768s3 /mnt/disk300
sirius / # df -h
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c0d512s3      2.2G   1.2G  1016M    55%    /
/devices                 0K     0K     0K     0%    /devices
/dev                     0K     0K     0K     0%    /dev
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   159M   452K   158M     0%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
fd                       0K     0K     0K     0%    /dev/fd
swap                   158M     0K   158M     0%    /tmp
swap                   158M    12K   158M     0%    /var/run
/dev/dsk/c0d426s3       66M   1.0M    58M     2%    /mnt/disk1aa
/dev/dsk/c0d768s3      274M   1.0M   246M     0%    /mnt/disk300
```

I tested ZFS a little too, but didn't much exercise it

Adding a disk works on the fly, too

```
 00: CP LINK * 200 F202
00: DASD F202 LINKED R/W
WARNING: Channel Report:
  Solicited:       0
  Overflow:        0
  Chain:           0
  Source Code:     03
  Ancilliary:      1
  Recovery Code:   04
  Source ID:       0001

NOTICE: Volume TD1200 discovered at 0f202 with blockize 4096 and offset 634

WARNING: New device f202 online
```

[Smart enough to “do the right thing” when a disk shows up]

Remove an (unused) disk

```
00: CP Q V DA
00: DASD 0190 3390 V53ZAR R/O    107 CYL   0213 RELN ON DASD  6E0B SUBCHANNEL =
00: DASD 0191 3390 VUSZA1 R/W     50 CYL   3080 RELN ON DASD  6E09 SUBCHANNEL =
00: DASD 019D 3390 ZVM530 R/O    115 CYL   1548 RELN ON DASD  44D6 SUBCHANNEL =
00: DASD 019E 3390 V53ZAR R/O    250 CYL   0935 RELN ON DASD  6E0B SUBCHANNEL =
00: DASD 01AA 3390 (TEMP) R/W    150 CYL   1670 RELN ON DASD  6E0C SUBCHANNEL =
00: DASD 0200 3390 VUSZA1 R/W   3338 CYL   3160 RELN ON DASD  6E09 SUBCHANNEL =
00: DASD 0201 3390 VUSZA2 R/W   3338 CYL   3339 RELN ON DASD  6E0A SUBCHANNEL =
00: DASD 0300 9336 (VDSK) R/W 600064 BLK   0000 RELN ON DASD  VDSK SUBCHANNEL =
00: DASD 0319 3390 VMPP02 R/O     75 CYL   0898 RELN ON DASD  4433 SUBCHANNEL =
00: DASD F200 3390 VUSZA2 R/W   3338 CYL   0001 RELN ON DASD  6E0A SUBCHANNEL =
00:
00: CP DET 201
00: DASD 0201 DETACHED
WARNING: Channel Report:
 Solicited:     0
 Overflow:      0
 Chain:         0
 Source Code:   03
 Ancilliary:    1
 Recovery Code: 04
 Source ID:     0002

WARNING: Device 0201 removed
```

Smart enough to “do the right thing” when a disk goes away

Add swap and use it

```
sirius / # mkfile 250m /mnt/diskf200/swap250m1
sirius / # swap -a /mnt/diskf200/swap250m1
operating system crash dump was previously disabled --
invoking dumpadm(1M) -d swap to select new dump device
sirius / # swap -l
swapfile             dev  swaplo blocks     free
/mnt/diskf200/swap250m1  -          8   511992   511992
sirius / # swap -s
total: 26844k bytes allocated + 68016k reserved = 94860k used, 360616k available
[ I generated some load ]
sirius / # swap -s
total: 54400k bytes allocated + 85080k reserved = 139480k used, 313856k available
sirius / # vmstat 30
 kthr      memory            page            disk          faults      cpu
 r b w   swap  free  re  mf pi po fr de sr 0f 00 00 rm   in   sy   cs us sy id
 0 0 0 196364 255532  0   0  0  0  0  0  0  3  0  4 -0    0    0    0  0 99  0
 0 0 0 248524 75496 523  95 1224 98 98 0 0  0  0 62  0 2012 2979  975 28  9 61
 0 0 0 204984 32520   0   0  0 94 94  0  0  0  0  7  0 1418  161  114 43  1 54
^C
```

A look from the outside (during iozone)

```
  CP IND USER ZIP00SOL
USERID=ZIP00SOL MACH=ESA STOR=512M VIRT=V XSTORE=NONE
IPLSYS=DEV 0191 DEVNUM=00017
PAGES: RES=00116045 WS=00116042 LOCKEDREAL=00000035 RESVD=00000000
NPREF=00000001 PREF=00000000 READS=00000122 WRITES=00000053
XSTORE=000000 READS=000000 WRITES=000000 MIGRATES=000000
CPU 00: CTIME=88:26 VTIME=094:57 TTIME=104:00 IO=786965
        RDR=000000 PRT=001438 PCH=000000 TYPE=CP   CPUAFFIN=ON
USERID=ZIP00SOL MACH=ESA STOR=512M VIRT=V XSTORE=NONE
IPLSYS=DEV NONE DEVNUM=00017
PAGES: RES=00116045 WS=00116042 LOCKEDREAL=00000035 RESVD=00000000
NPREF=00000001 PREF=00000000 READS=00000122 WRITES=00000053
XSTORE=000000 READS=000000 WRITES=000000 MIGRATES=000000
CPU 01: CTIME=88:25 VTIME=088:43 TTIME=097:26 IO=783035
        RDR=000000 PRT=000000 PCH=000000 TYPE=CP   CPUAFFIN=ON
 CP IND
AVGPROC-011% 02
XSTORE-000000/SEC MIGRATE-0000/SEC
MDC READS-000000/SEC WRITES-000000/SEC HIT RATIO-000%
PAGING-0/SEC STEAL-000% SPOOL-019%
Q0-00001(00000)                           DORMANT-00026
Q1-00000(00000)           E1-00000(00000)
Q2-00000(00000) EXPAN-001 E2-00000(00000)
Q3-00004(00000) EXPAN-001 E3-00000(00000)
PROC 0000-011% CP       PROC 0001-011% CP
LIMITED-00000
```

Other things I've tested without incident

- Typical Unix user and admin commands
  - Didn't get around to NIS or LDAP yet
- Basic networking
  - NFS client, FTP, scp/ssh
  - Ifconfig, netstat
- Least Privileges / Role Based Access Control (RBAC) privilege bracketing
- C compiler (gcc), make, configure, related tools
- IEEE FP conformance (paranoia.c)
- lighthttpd (I had to ln -s /usr/local/lib/libpcre.so.0.0.1 /lib/libpcre.so.0)

Other Experiences

- By necessity, the following slides describe errata
- Nobody should take umbrage at this
  - > This is a great accomplishment, and it would be a miracle if it approached functional completeness
  - > Sun is experienced with complete ports of the OS, and we know it takes <u>massive</u> effort and resources
  - > Any missing feature or defect can be rectified by appropriate commitment of time, money, and effort: many engineers, testers, doc writers, and $millions.

Errors as of September

October build fixed several

```
Directory creation error on console if connecting with ssh -X
hostid command does not work
zonecfg dumps core [zones haven't been tested]
prstat command missing [fixed in current build. yay!]
man command missing [fixed in current build]
Dtrace not implemented
format reports 'no disks found'
kstat and psrinfo fail
prtconf -vp gives bogus error
Apache missing [is listed in doc but isn't present]
package maintenance commands? [needs patch/package utilities]
psradm -n 1 hangs system (goes into loop)
Need a /etc/release file to identify s/w level
digest -a md5 command / PKCS failure [closed source issue?]
telnet enabled by default [should be 'netservices limited' - fixed]
elfsign command fails [nothing to do with Galadriel]
fmadm and other fma commands are missing
pfiles command, several failure modes including failure of target proce
missing 'tr' command in expected PATH
pstack command dumps core
'/usr/bin/getconf -a' dumps core [fixed in current build]
64-bit version of ls fails [fixed in current build]
No NFS server capability [requires kernel lock manager]
```

Other missing features (some may be in October build)

- No DTrace – an advanced feature of Solaris 10
  - > No surprise: ported to BSD and Mac, but probably has processor dependent code to be dealt with for z
- No prstat command
  - > Only 'top' which is non-standard, doesn't report as much info, and changes the system you're looking at
- No 'project' facility, Fair Share Scheduler, rcapadm, dynamic resource pools (poolcfg/pooladm)

Service/patching

- No pkg* utilities so you can't add, remove packages
- No Live Upgrade or even standard upgrade. No smpatch. Can't identify service level
- The only way to upgrade the system so far is complete DDR image restore from install media, after having backed up any change you've made
  - Host and network identity, userids
  - Any software you've installed
- Remember that these are early days; surely this will be addressed

Errata in a different way...

- Built Hercules on z for fun, and as a good exercise
  - Maybe run VM/370 or CentOS under Hercules under OpenSolaris under z/VM... or z/VM itself
  - One of my personal performance benchmarks is “build Hercules, then get MIPS counts”
- When launched, provoked a CP ABEND HTT001
- Decided to not try this again for a while, as that might make me an unwelcome guest
- Incident open with IBM L2. They asked us to recreate this but it didn't crash when we wanted it

Performance – and note the caveats!

- Intense interest in platform comparisons
- Results here are not “formal” benchmarks.
  - I only have access to limited z, SPARC, and x86 configs
  - Can't really test I/O performance with only a few disks
  - Can't really test general performance without apps to use
- So, these results are my idiosyncratic tests – I consider them “evocative” and “illustrative” :-)
- Sirius was compiled without optimization and with debugging turned on. Expect an optimized version to be faster. But: all binaries I built, in user space, had optimization turned on.

Performance – iperf benchmark

iperf	**Mbits/sec**	**Notes**
	higher is better	
OpenSolaris on z (sirius)	791	
UltraSPARC T1 1GHz	2060	
UltraSPARC-III at 1.5GHz	3330	
SPARC64 VI at 2.15GHz	5390	
z/Linux	2140	SLES10 on same z9
Following are inter-guest on the z9:		
z/Linux to sirius	1.7	SLES is iperf client
sirius to zlinux	40.4	sirius is iperf client

Raw network traffic with client and server in same OS
except for sirius<-->z/Linux guests on same z/VM instance
Note: sirius network stack is early and unoptimized
SLES 10 result indicates potential improvement
No idea why the numbers for sirius <--> z/Linux are asymmetric

Performance – building Hercules

	CPU	Elapsed
sirius on z9	16m42s	18m43s
t5240 1.4GHz	22m36s	22m18s
M-Series SPARC64	5m21s	5m20s
dual AMD 2.7GHz	6m52s	6m53s
UltraSPARC IIIi 1.5GHz	9m55s	10m01s
UltraSPARC IIIi 1.2GHz	13m35s	13m41s

CPU is sum of user + system CPU times
man time' says 'real' (elapsed) time can be less than CPU on multiprocessor machines
Note: the t5240 is using 1/128th of the machine
Compiling Hercules is a pretty compute intensive application on each platform
Compiles measure integer and character manipulation and function calls
Remember: gcc on sirius is not optimized, so the result might improve a lot

Performance – Hercules MIPS

MIPS by instruction	**A**	**B**	**BCTR**	**L**	**LA**	**SIEVE**
sirius on z9	11.7	25.1	21.3	14.1	25.4	24.8
AMD 2.7GHz	46.2	78.3	78.7	52.1	83.7	78.3
M-series SPARC64	20.7	40.9	40.4	25.3	42.1	41.4
t5240 SPARC 1.4GHz	3.4	5.9	5.6	3.7	6.2	5.9
SUSE on z9	9.7	22.3	16.8	10.6	19.5	21.9

I have found Hercules simulated MIPS rates a good test for CPU performance
Exercises integer arithmetic, branching, calls, memory latency
(all are instruction kernels, except SIEVE – prime number program)
Note: the t5240 is using 1 CPU of 128 in 2RU – we could run 128 of these with little degradation

No idea why the SuSE and Sirius numbers differ: both gcc -O3

Isn't it amazing that you can SIMULATE the performance of a 9021-711 on your desktop?

Performance – iozone benchmark

iozone	Writes	Reads	Elapsed	CPU
	MB/sec	MB/Sec		
OpenSolaris on z9	16.3	26.9	97m49s	15m03s
US-II 440Mhz	15.1	16.2	151m25s	38m09s
Pentium III 1GHz	68.5	101.7	34m47s	19m41s
T5240 CMT 1.4GHz	79.4	591.6	23m07s	7m39s
SLES 10 on z9	17.6	17.3	128m13s	4m11s

Sequential I/O with file sizes big enough to ensure actual disk I/O, not just cache

(At least for writes. I'm willing to believe the 5240 results reflect cache)

For laughs: the US-II is my 1999 Ultra10 workstation, and the P3 is an old PC

Unfortunately, I didn't have access to the SPARC M-series for this test

Based on speed comparison to T5240, would have had CPU under 2 minutes

In any case, this is disk I/O bound, not CPU bound, on all systems.

SLES used little CPU, but had some poor throughput numbers, esp. reads, hence elapsed time significantly higher than sirius

Performance - Linpack

	kFLOPS	Compile time (seconds)
sirius on z9	172538	2.7
M-series SPARC64	883513	0.75
AMD 2.7GHz	878427	0.45
UltraSPARC IIIi 1.2GHz	216903	1.6
SLES 10 on z9	180450	

Remember that gcc on sirius is not optimized
But the compiled output is, on all platforms.
SLES 10 on z9 Kflops pretty close to that for sirius. I forgot to time gcc
gcc does not provide best SPARC results – Sun's compiler would yield better

Performance summary

- A z9 is slower than UltraSPARC III at 1.5GHz
  - (processor in small, low-cost, back-level SPARC boxes)
- Recent (2007) SPARC 2-6 times faster
  - I didn't try our faster 2008 models
- Even for disk I/O (we passed 18GB/s, years ago)
- All tests run on an idle z9 and near-idle Sun boxes
- *Not* rigorous, but consistent with my work experience
- Optimized sirius kernel code and network stack should be substantially faster
  - So would SPARC apps built with Sun's own compiler (gcc for SPARC notoriously non-optimal)

Performance suggestion

- I've expressed caveats about the limited testing I was able to do and limits on its applicability
  - Infer, but understand the limits
  - I would be delighted to see “proper” z benchmarks
- Sun reports performance results for its products
  - http://www.sun.com/benchmarks/
- We think that's an essential part of open systems
- If a vendor claims they can run certain workloads, they should prove it - out in the open
  - If you think that's important, then contact your systems vendor and insist on it

Evaluation and perspective

- This is impressive, even historic, engineering by SNA – in particular by Neale Ferguson, who did the lion's share of the work. Show respect, folks.
- Shows that a Unix operating system can be ported to another platform
  - This is not a surprise, right?
- At this moment – and this is early days – the implementation has significant functional gaps
- Substantial effort is required – and a community to do it – if this is to be a full and compatible implementation

What does Sun think about it?

- We really like the idea of Solaris being open and everywhere. That's why we made OpenSolaris
- We don't see sirius as a revenue producer for Sun
  - > You don't make money by encouraging people to buy products (servers, services) from someone else
  - > Solaris already competes against Linux on x86 (doesn't require another platform to enable direct comparison)
  - > We believe value comes from volume
- We don't see this as significant for Sun STK
  - > You don't make platform decisions as a response to FUD
  - > Customer loyalty to STK products is based on products and service, just as every vendor's products

It's in the open now

- The effort should be in the open with an OpenSolaris.org community, with code developed in the open and with feedback from interested parties.
  - Stephen Harpster, Sun, on 9/6/2005: “*First thing, go to http://www.opensolaris.org. Click on the "register" link in the upper right corner. You need to create an account in order to start a discussion board. Next, login with your new account name and go to http://www.opensolaris.org/jive/forum.jspa?forumID=13. Click on Post New Thread and say you want to create a new community for porting OpenSolaris on zSeries hardware. That's it!*”
- That's the past - now the source is out :-)

Platform requirements

- The implementation requires z9 or later, and z/VM 5.3 or later (with VM64466/UM32414)
  - More restrictive than z/Linux, which can run on Hercules as well as on older z-kit
  - Maybe my recommending “as VM guest only” a mistake in that sense, but I expect it made the port effort much more tractable. Besides, I (heart) VM
  - Perhaps Hercules can be extended to provide the appropriate DIAGNOSE codes, and OpenSolaris rebuilt without -march=z9-109

What will be required for success?

- First, define “success”
- If “a full implementation suitable for migration”
  - millions of dollars will have to be spent to ensure completeness and correctness of implementation, documentation, service and support
  - ISV support: IBM (Java, WAS, DB2, etc), Oracle, ...
  - Frankly, there are more viable, cost-effective ways, with smaller obstacles to adoption and better price/performance
- If success is defined as “alternative to Linux on z for targeted purposes”, then I think quite reasonable and desirable

What will be required for success?

- A community is essential
  - “Given enough eyeballs, all bugs are shallow”. This project needs more eyeballs than have access to z9/z10
  - So, it will have to run under Hercules, and new community participants must have their say on implementation. Enterprising souls may some day:
    - Build OpenSolaris on z without -march=z9-109
    - Modify Hercules to provide needed DIAG interfaces, or
    - Modify OpenSolaris on z to use CTCA, SSCH, not OSA, DIAG
    - Port applications, provide 3rd party support
- Requires enthusiasm from the Linux and Solaris communities

Good news: plenty of work for everyone!

- Full implementation: if there's no DTrace or Solaris Containers, or... then it's not Solaris
- Use ZFS as boot file system, and use the new Image Packaging System (IPS)
  - > Repository based patch and update management, rollback, undo, clone...
  - > This is how Solaris will be maintained in future and is already how OpenSolaris works
- Open source application stacks:
  - > AMP (Apache, MySQL, Perl/PHP/Python)
  - > MARS (MySQL, Apache, Ruby, Solaris)

Get involved

- Learn Solaris. There's a lot there that is absolutely outstanding. This is a true enterprise OS
  - > A free download, or ask, and I'll get you a CD
  - > It will run on your PC, natively or under VMware or xVM
- Join the community – that's “the way” in open source
  - > Contribute: document, test, comment, code
- Note these URLs:
  - > http://www.opensolaris.org/os/project/systemz/
  - > http://distribution.sinenomine.net/opensolaris
- Join SOL-390 at vm.marist.edu

Thank you!

jeff.savit@sun.com