The 2.6 Linux Kernel and Red Hat Enterprise Linux

August 2004

Agenda

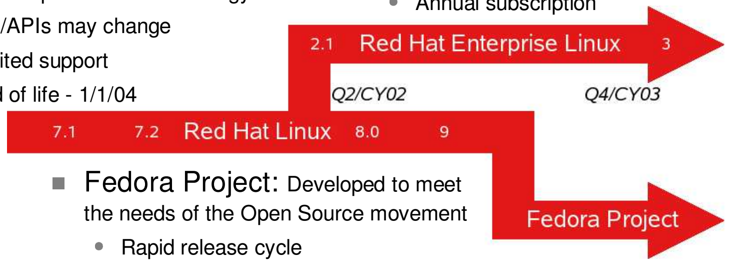
- Overview of the Red Hat Enterprise Linux family
- Summary comparison of features included in the Linux 2.6 kernel and Red Hat Enterprise Linux 3
- A technical look at the operation and benefits of some primary Linux 2.6 kernel features that are included in Red Hat Enterprise Linux 3
- Features exclusive to Red Hat Enterprise Linux 3
- Questions

Agenda

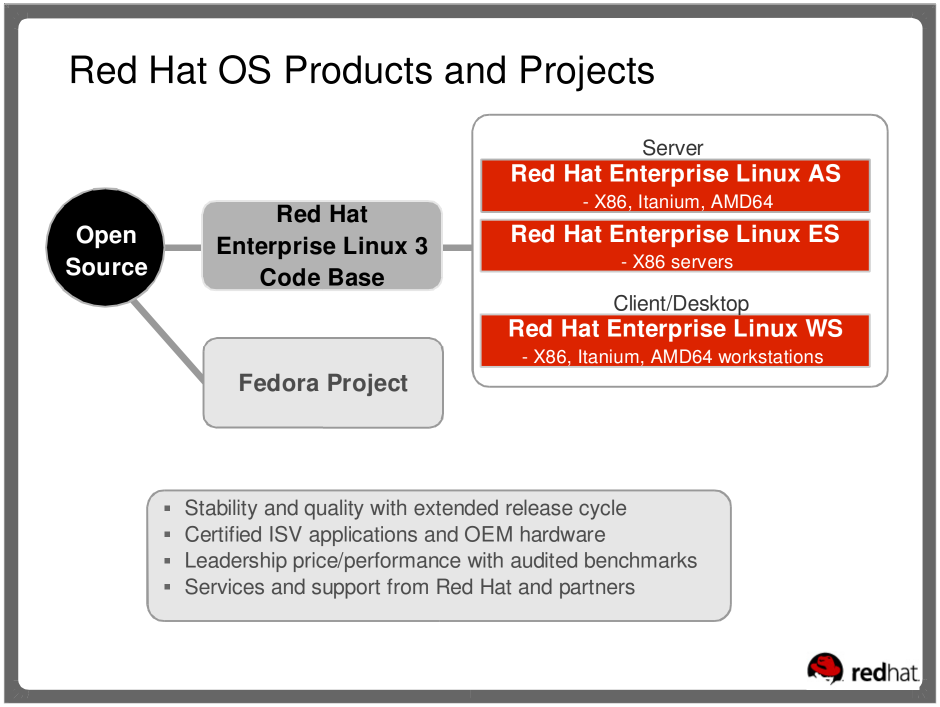
- Overview of the Red Hat Enterprise Linux family
- Summary comparison of features included in the Linux 2.6 kernel and Red Hat Enterprise Linux 3
- A technical look at the operation and benefits of some primary Linux 2.6 kernel features that are included in Red Hat Enterprise Linux 3
- Features exclusive to Red Hat Enterprise Linux 3
- Questions

Red Hat Product Lineage

- Red Hat Linux: Developed to meet the needs of the Open Source movement and early technology adopters
  - 4-6 month release cycle
  - Latest open source technology
  - ABI/APIs may change
  - Limited support
  - End of life - 1/1/04
- Red Hat Enterprise Linux: Developed to meet the needs of enterprise/commercial customers
  - 12-18 month release cycle
  - Stable/mature open source technology
  - ABI/APIs held stable
  - Bundled support – up to 5 years
  - Annual subscription



- Fedora Project: Developed to meet the needs of the Open Source movement
  - Rapid release cycle
  - Latest open source technology
  - ABI/APIs may change
  - No support; free download



Red Hat OS Products and Projects

Open Source

Red Hat Enterprise Linux 3 Code Base

Fedora Project

Server

Red Hat Enterprise Linux AS
- X86, Itanium, AMD64

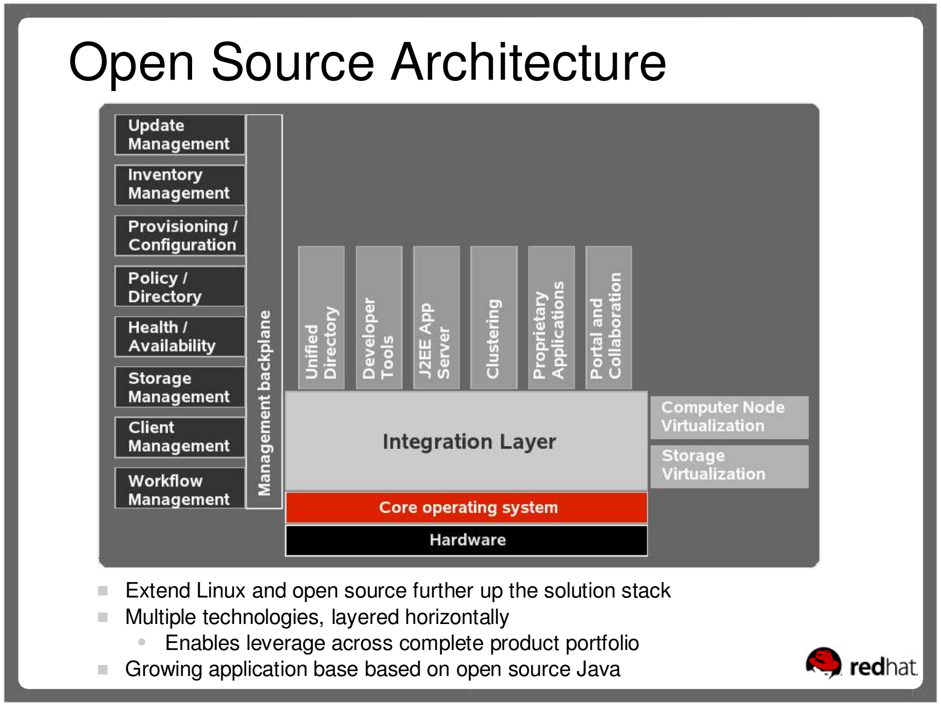
Red Hat Enterprise Linux ES
- X86 servers

Client/Desktop

Red Hat Enterprise Linux WS
- X86, Itanium, AMD64 workstations

- Stability and quality with extended release cycle
- Certified ISV applications and OEM hardware
- Leadership price/performance with audited benchmarks
- Services and support from Red Hat and partners

redhat

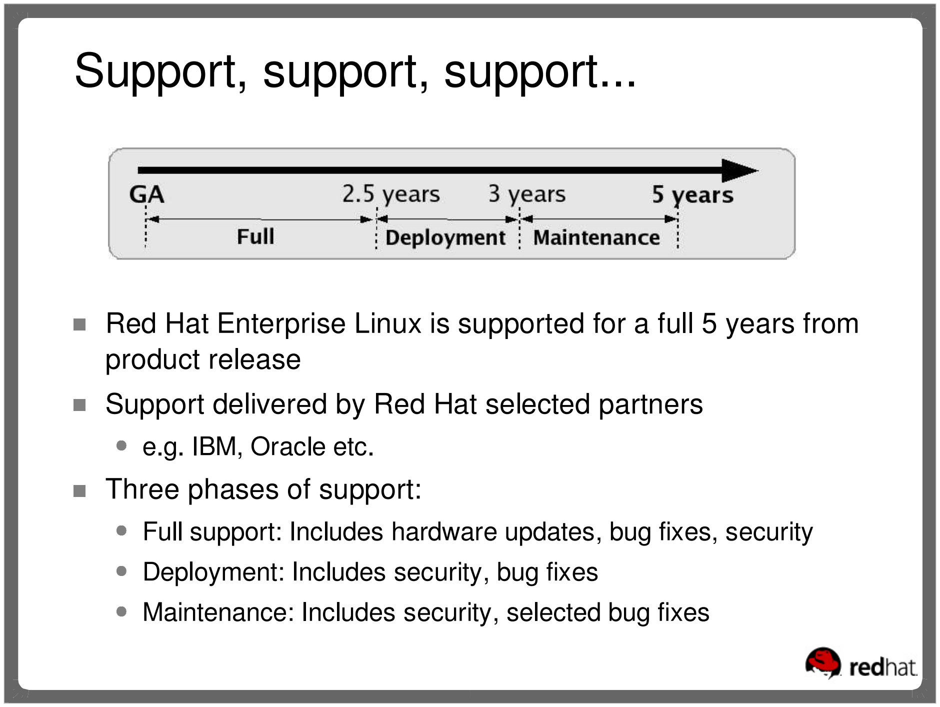


Open Source Architecture

Update Management
Inventory Management
Provisioning / Configuration
Policy / Directory
Health / Availability
Storage Management
Client Management
Workflow Management
Management backplane
Unified Directory
Developer Tools
J2EE App Server
Clustering
Proprietary Applications
Portal and Collaboration
Integration Layer
Computer Node Virtualization
Storage Virtualization
Core operating system
Hardware

- Extend Linux and open source further up the solution stack
- Multiple technologies, layered horizontally
  - Enables leverage across complete product portfolio
- Growing application base based on open source Java

redhat

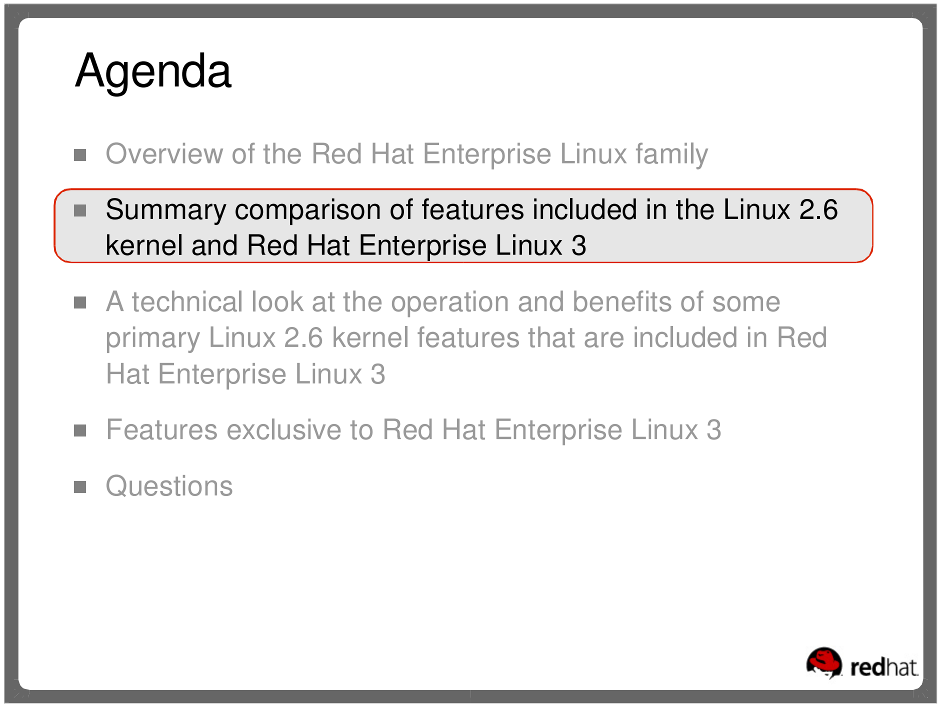


Support, support, support...

GA
2.5 years
3 years
5 years
Full
Deployment
Maintenance

- Red Hat Enterprise Linux is supported for a full 5 years from product release
- Support delivered by Red Hat selected partners
  - e.g. IBM, Oracle etc.
- Three phases of support:
  - Full support: Includes hardware updates, bug fixes, security
  - Deployment: Includes security, bug fixes
  - Maintenance: Includes security, selected bug fixes

redhat



Agenda

- Overview of the Red Hat Enterprise Linux family
- Summary comparison of features included in the Linux 2.6 kernel and Red Hat Enterprise Linux 3
- A technical look at the operation and benefits of some primary Linux 2.6 kernel features that are included in Red Hat Enterprise Linux 3
- Features exclusive to Red Hat Enterprise Linux 3
- Questions

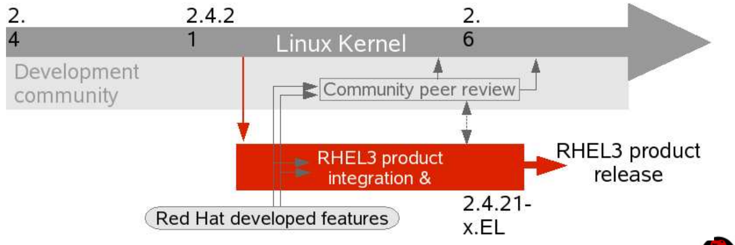
redhat

Linux 2.6 & RHEL 3 Kernel Features

- Red Hat Enterprise Linux 3 is based on a hybrid kernel, comprising:
  - 2.4.21 basic core
  - Numerous “backported” features from the 2.6 kernel
  - Additional features not yet included in the 2.6 kernel
- Overall goal is to provide exceptional stability combined with the technical features required by customers and ISVs
- The few 2.6 features that are not included are:
  - Still insufficiently stable for commercial/enterprise use
  - Not urgently needed at this time

Upstream?

- Linux community kernel versions are always ahead of commercial Linux product kernels
  - Productization cycles introduce version skew
- Features that Red Hat and others develop, which appear in “*Upstream”* kernels, are often integrated into Red Hat Enterprise Linux products that are based on earlier kernels
  - Especially features suitable for commercial environments



Linux 2.6 & Enterprise Linux 3 Kernel Features

Feature	Included in Linux 2.6 kernel	Included in RHEL 3 products
Native Posix Thread Library (NPTL)	Yes	Yes
Kernel IPSec	Yes	Yes
Asynchronous I/O (AIO)	Yes	Yes
O(1) Scheduler	Yes	Yes
Oprofile	Yes	Yes
Kksymoops	Yes	Yes
Reverse Map Virtual Memory (rmap VM)	Yes	Yes
HugeTLBFS	Yes	Yes
Remap_file_pages	Yes	Yes
IGMPv3	Yes	Yes
Ipvs	Yes	Yes
Access Control Lists (ACLs)	Yes	Yes
4GB-4GB memory split	No	Yes
Scheduler support for hyperthreaded CPUs	No	Yes
Block I/O (BIO) layer	Yes	No
Support for >2 TB file system	Yes	No
New I/O elevators	Yes	No

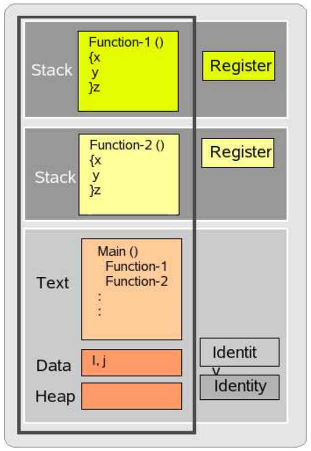
- Enterprise Linux 3.0 includes majority of the performance enhancements found in 2.6 kernel.
- Not the kernel itself, but all the components of the kernel that are valuable. RH ships all enterprise-ready, performance enhancing components of the 2.6 kernel.
- **For more details, check out White Paper available at www.redhat.com**

Agenda

- Overview of the Red Hat Enterprise Linux family
- Summary comparison of features included in the Linux 2.6 kernel and Red Hat Enterprise Linux 3
- A technical look at the operation and benefits of some primary Linux 2.6 kernel features that are included in Red Hat Enterprise Linux 3
- Features exclusive to Red Hat Enterprise Linux 3
- Questions

Native Posix Thread Library

- Required for high performance multi-threaded commercial applications, e.g. Java
- Full implementation of POSIX threads
- Major feature that will accelerate Linux adoption in the enterprise
- Highly scalable, native implementation
  - Creation/deletion performance independent of the number of threads running
  - Includes threaded core dumps
  - Informal benchmarks show >50,000 simultaneous thread creations-deletions/second
- Thread Local Storage & Futex APIs

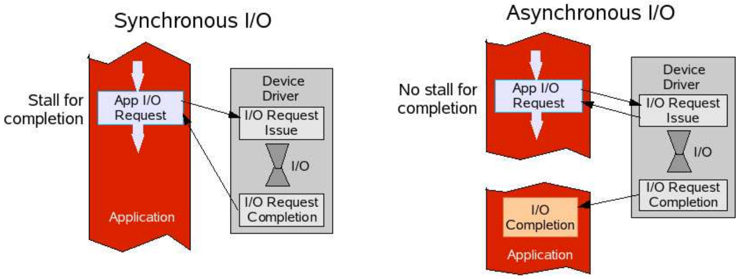


Benchmark use of NPTL

- Clients ran Apache 2.0 with the worker MPM model
  - Worker MPM uses a combination of processes and threads
  - Each client had 33 Apache server processes
  - Each Apache process had 500 threads
  - Supported16,500 users on a client
  - The Apache threads used NPTL
- NPTL allows a benchmark to run more users on each client which reduces the total number of clients
  - Fewer clients means better price/performance

Asynchronous I/O

- Allows application to continue processing while I/O is in progress
  - Eliminates Synchonous I/O stall
- Critical for I/O intensive server application
  - e.g Database writer deamons
- Red Hat Enterprise Linux feature since v2.1, May 2002


Synchronous I/O

Asynchronous I/O

Benchmark use of Qlogic Driver

- The Qlogic driver contained in RHEL3 has been performance optimized by Red Hat
- Red Hat added a ql2xintrdelaytimer module parameter to the Qlogic driver that affects interrupt frequency.
  - Value is in units of 100 usecs
  - Default in RHEL 3 is 3
  - Increase value to reduce interrupts and save cpu cycles.
    - IO latencies will be higher
    - Better for cpu intensive workloads

```
/sbin/insmod qla2300 ql2xintrdelaytimer=10
```

  - Decrease value to increase interrupts and decrease latency.
    - More cpu consumed by interrupt processing
    - Better for throughput sensitive workloads

```
/sbin/insmod qla2300 ql2xintrdelaytimer=0
```

Benchmark use of Direct I/O

- Red Hat Enterprise Linux 3 supports the use of the O_DIRECT flag on block devices
  - Same performance as raw access
  - Don’t have to worry about the 255 raw device limit
    - Very important for large RAC configurations
- Quick and direct access to the IO devices is very important for database performance
  - O_DIRECT bypasses the kernel buffer caching
  - Kernel buffer cache is inefficient for database data
    - The database already has a copy of the data in it’s own cache.
    - No need to cache it twice
- O_DIRECT can be used to optimize other IO intensive programs such as dd

Changing Interrupt Affinity

- Check the interrupt affinity scheme

```
cat /proc/interrupts
```

- Change the interrupt affinity it is not optimal for your setup
  - Change the smp_affinity value to the bitmask value for the processor you want the interrupts on
  - E.g.: Affinity interrupts for the device with IRQ60 to processor 6

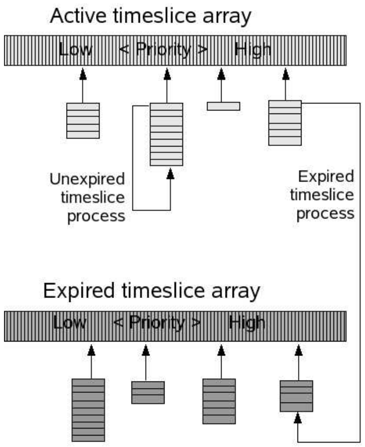
```
echo 00000020 > /proc/irq/60/smp_affinity
```

- Here, ethx bound to cpu0; qla2300 bound to cpu1

```
# cat /proc/interrupts
           CPU0       CPU1       CPU2       CPU3
  31:         0          0          0          0        LSAPIC  cmc_hndlr
  . . . .
  56:         0         30          0          0  IO-SAPIC-level  sym53c8xx
  57:   3724164          0       1017          0  IO-SAPIC-level  eth0
  58:         0   12663646          0       6444  IO-SAPIC-level  qla2300
  59:      7289   12668058          0          0  IO-SAPIC-level  qla2300
  60:         0    4572587          0          0  IO-SAPIC-level  qla2300
  61:   3492189          0        804          0  IO-SAPIC-level  eth1
  62:         0   12200095          0       7170  IO-SAPIC-level  qla2300
  63:   3493641          0          0          0  IO-SAPIC-level  eth2
```

O(1) Scheduler

- Highly scalable scheduler that chooses the next process to run in a constant time
  - Low overhead
- Scales well
  - With processor count
  - With process count
- Separate queue maintained for all processes at a given priority
  - Timeslice allocated based on priority & I/O activity
- Uses 2 process priority arrays
  - Active & expired timeslice
- Arrays are swapped when all timeslices have expired



Benchmark use of Scheduler

- Benchmarks carefully tune process priority and CPU affinity
- For best Oracle performance, raise the priority of the Oracle processes
  - Nice, renice, or sched_setscheduler()
  - Make the log writer process a little higher than the other Oracle processes.
- In certain situations it is useful to override the scheduler and affinitize a process to a particular processor
  - Use the taskset command
  - Examples
    - Affinity process 2157 to cpu 2

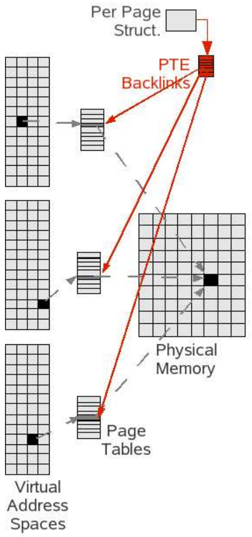
```
taskset 0x00000004 –p 2157
```

    - Affinity process 2157 to cpus 0 and 1

```
taskset 0x00000003 –p 2157
```

Reverse Map VM

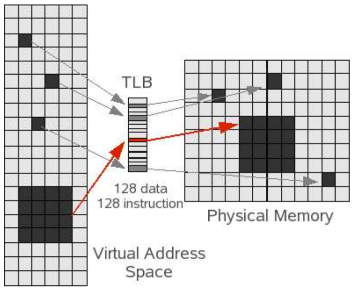
- Page Tables are used to translate a virtual address to a physical memory address
  - A single physical page can be mapped by several virtual address spaces
- Before the 2.6 kernel, finding out which virtual addresses spaces were mapping a physical page was very hard
  - Required a full scan of all page tables
- In 2.6/RHEL3 the per-page struct includes pointers to all PTEs that map the page
  - Eliminates page table scan – reduces the time to perform page out operations
- Greatly improves performance for:
  - Memory constrained systems
  - NUMA memory systems
  - Systems with large aggregate virtual address spaces



redhat

HugeTLBFS

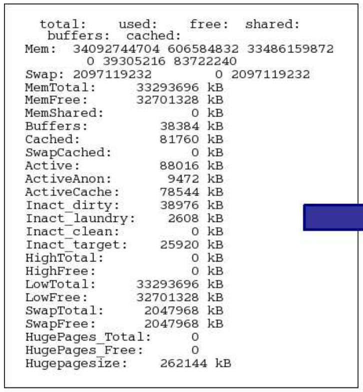
- The Translation Lookaside Buffer (TLB) is a small CPU cache of recently used virtual to physical address mappings
- TLB cache misses are extremely expensive on today's very fast, highly pipelined CPUs
- Large memory applications can incur high TLB miss rates
- HugeTLBs permit memory to be managed in very large segments
  - E.G. Itanium:
    - Standard page: 16KB
    - Default huge page: 256MB
    - 16000:1 difference
- File system mapping interface
- Ideal for databases
  - E.G. TLB can fully map a 32GB Oracle SGA



redhat

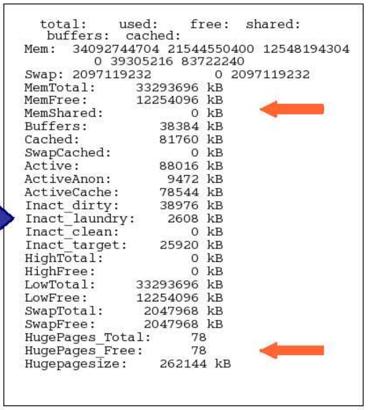
Allocating HugeTLB pages

- Declare size in /proc
  - E.G.: echo 20000 > /proc/sys/vm/hugetlb_pool



```
  total:    used:    free:  shared:
   buffers:  cached:
Mem:  34092744704 606584832 33486159872
        0 39305216 83722240
Swap: 2097119232         0 2097119232
MemTotal:     33293696 kB
MemFree:      32701328 kB
MemShared:           0 kB
Buffers:         38384 kB
Cached:          81760 kB
SwapCached:          0 kB
Active:          88016 kB
ActiveAnon:       9472 kB
ActiveCache:     78544 kB
Inact_dirty:     38976 kB
Inact_laundry:    2608 kB
Inact_clean:         0 kB
Inact_target:    25920 kB
HighTotal:           0 kB
HighFree:            0 kB
LowTotal:     33293696 kB
LowFree:      32701328 kB
SwapTotal:     2047968 kB
SwapFree:      2047968 kB
HugePages_Total:     0
HugePages_Free:      0
Hugepagesize:    262144 kB
```

Before



```
  total:    used:    free:  shared:
   buffers:  cached:
Mem:  34092744704 21544550400 12548194304
        0 39305216 83722240
Swap: 2097119232         0 2097119232
MemTotal:     33293696 kB
MemFree:      12254096 kB
MemShared:           0 kB
Buffers:         38384 kB
Cached:          81760 kB
SwapCached:          0 kB
Active:          88016 kB
ActiveAnon:       9472 kB
ActiveCache:     78544 kB
Inact_dirty:     38976 kB
Inact_laundry:    2608 kB
Inact_clean:         0 kB
Inact_target:    25920 kB
HighTotal:           0 kB
HighFree:            0 kB
LowTotal:     33293696 kB
LowFree:      12254096 kB
SwapTotal:     2047968 kB
SwapFree:      2047968 kB
HugePages_Total:    78
HugePages_Free:     78
Hugepagesize:    262144 kB
```

After

redhat

Allocating HugeTLB pages

- The hugetlb page memory pool must be physically contiguous memory pages.
  - The longer the system has been booted the more likely the kernel won't be able to find enough contiguous memory
  - Allocate the hugetlb pool early
  - Can even allocate the hugetlb page pool as the kernel boots
    - Best chance of getting contiguous memory
    - Add an option to the kernel boot line

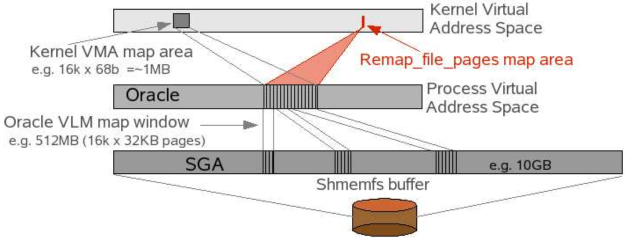
  "hugetlbpool=20000"   (size expressed in MB)
- The pool of hugetlb pages may be resized dynamically

  echo 0 > /proc/sys/vm/hugetlb_pool

redhat

Remap_file_pages

- Shmemfs mapping enhancement for x86 (32 bit) systems
  - Useful for large shmemfs apps. e.g. Oracle VLM
- Shmemfs buffers consume kernel space mapping structures
  - e.g. 512MB buffer requires ~1MB/process kernel structure space
  - So 800 Oracle processes (SQL commands) would consume most available kernel space
- Shrinks 1MB of structures to a single small structure (<100b)
  - Allows many more concurrent SQL commands


Kernel Virtual Address Space
Kernel VMA map area
e.g. 16k x 68b =~1MB
Remap_file_pages map area
Oracle
Process Virtual Address Space
Oracle VLM map window
e.g. 512MB (16k x 32KB pages)
SGA
e.g. 10GB
Shmemfs buffer

Oprofile

- Code profiling support included in the kernel – Oprofile
  - System-wide profiler daemon, capable of profiling multiple events, in any kernel/library/application code
  - Uses hardware performance counters in the CPU
  - Includes several post-profiling tools
- Helps application designers identify:
  - Loop unrolling; poor cache utilization; inefficient type conversion; branch mispredictions; etc
- Visit http://oprofile.sourceforge.net

Networking

- Improvements to channel bonding
  - Failover & bandwidth aggregation for servers w/multiple NICs
- Kernel IPsec – secures IPv4 traffic
  - Tunnel mode builds tunnels between subnets
  - Transport mode secures communication directly between two machines
  - Packets are encrypted, authenticated and anti-replay protected
  - Able to communicate with IPsec devices and OS
- Kernel IPv6 support (more complete implementation than in 2.1)
- Kernel support for both IGMP V2 & V3 (Internet group management protocol)

NFS

- Significantly improved stability
- Client-side focused performance enhancements
  - NFSv3 readdirplus aggressively caches directory information
    - Improved file browsing; improved wire efficiency
- NFS over TCP
  - Suitable for congested networks
  - Server listens for TCP connections by default (& UDP)
  - Provides improved robustness vs. UDP
  - Client initiates TCP connection with mount -o tcp
- O_Direct support added
  - Client I/O bypasses buffer cache
  - Suitable for database applications
  - Parameter passed on open()

Access Control Lists

- File system ACLs
  - Unix file permissions not always adequate
    - Multiple UIDs, Groups, and set-UID apps proliferate
  - ACLs are additional sets of read/write/execute triplets
  - Can be added to any objects
    - Files, directories, devices, or any other file system objects
  - Highly configurable – fine tune access
    - Without resorting to multiple groups or set-UID apps
  - Includes support for NFS mounted file systems

Kksymoops

- Debugging feature that greatly simplifies kernel problem resolution by producing symbolic oops (crash) reports
  - Includes full kernel symbol table, not just module symbols
- Improves the quality of kernel bug reports
  - Aids rapid problem resolution
- Obvious feature for a supported Enterprise product

```
kernel BUG at time.c:100!
invalid operand: 0000
CPU:    1
EIP:    0060:[]    Not tainted
EFLAGS: 00010246
EIP is at sys_gettimeofday+0x84/0x90
eax: 0000004e   ebx: cee10000   ecx: 00000000   edx: 00000068
esi: 00000000   edi: 00000000   ebp: bffffad8   esp: cee11fa0
ds: 0068   es: 0068   ss: 0068
Process gettimeofday (pid: 566, threadinfo=cee10000 task=cf5b58a0)
Stack: 4001695c bffff414 40156154 00000004 c0112b20 cee10000 400168e4 bffffb44
       c0107973 00000000 00000000 40156154 400168e4 bffffb44 bffffad8 0000004e
       0000002b 0000002b 0000004e 400cecc1 00000023 00000246 bffffacc 0000002b
Call Trace:
 [] do_page_fault+0x0/0x49e
 [] syscall_call+0x7/0xb
```

Agenda

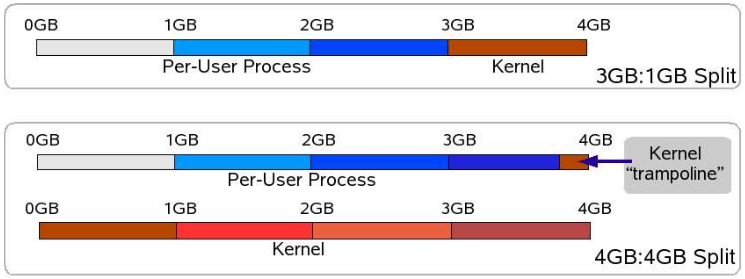
- Overview of the Red Hat Enterprise Linux family
- Summary comparison of features included in the Linux 2.6 kernel and Red Hat Enterprise Linux 3
- A technical look at the operation and benefits of some primary Linux 2.6 kernel features that are included in Red Hat Enterprise Linux 3
- Features exclusive to Red Hat Enterprise Linux 3
- Questions

4GB-4GB Split

- Major new capability to support large physical memories and increased application virtual address space
  - Enables practical support for very large physical memory configurations
    - 64GB in Enterprise Linux 3
  - Application virtual address space increased ~30% to almost 4GB
    - Enables support for larger user applications
  - This feature is only for the X86 architecture only
    - Not required for 64-bit architectures
  - Performance tests show minimal performance impact imposed by the additional memory management overhead
  - Included in the *hugemem* kernel

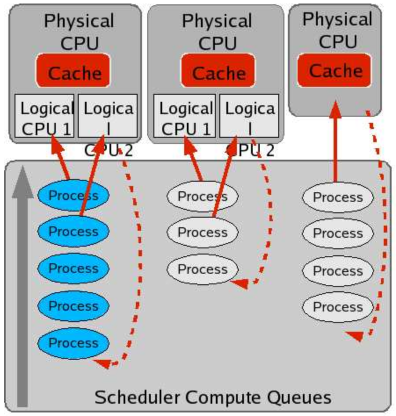
4GB-4GB Split

- A classic 32-bit 4GB virtual address space is split 3GB for user processes and 1GB for the kernel
- The new scheme permits 4GB of virtual address space for the kernel and almost 4GB for each user process



Hyperthreading CPU Scheduler

- Recognizes differences between logical and physical processors
- Optimizes process scheduling to take advantage of shared on-chip cache
- Implements one run queue per physical processor (as opposed to one run queue per processor or per system)
  - Strong CPU affinity avoids task bouncing



Crash dump

- Red Hat Enterprise Linux 3 includes Netdump
  - Network based kernel crash dump facility
- Utilizes Netdump server as a sink node for client/remote crash dumps and kernel log files (Netconsole)
- Simplifies crash analysis for Red Hat support/engineering groups
  - Reduces time to fix; increases system availability
- Currently dumps all memory
  - Enhancements in development include compression and memory type selection (e.g. free pages, user pages, page cache pages may be restricted)
- Dump can be analysed with Red Hat's crash utility
  - Significant additional kernel-specific capabilities layered on gdb
  - Can be run on a live system

Summary & Questions

- For commercial application deployments today, Red Hat Enterprise Linux 3 includes all the most important Linux 2.6 kernel technologies
  - Stable; supported; extensively-certified; benchmark proven
- The 4GB-4GB split in Red Hat Enterprise Linux 3 is particularly useful for large x86 systems
- Red Hat is working closely with its ISV & OEM partners to deliver the next release of Red Hat Enterprise Linux, based on the Linux 2.6 kernel, to customers
  - Alpha versions with partners today; Betas starting in Summer; final product targeted for Q1CY05
  - Fedora Core 2, based on Linux 2.6, due May 2004

IBM – Red Hat Product Certification

- IBM eServer System Certification
  - Most IBM eServers are Red Hat certified today
  - Specific models can be searched for at: http://hardware.redhat.com
- IBM Software Group
  - Red Hat Enterprise Linux is a Tier 1 operating system platform
  - IBM Software Group is committed to having all middleware and infrastructure applications certified for RHEL as quickly as possible upon general availability
- >1,000 Applications Certified with Red Hat Enterprise Linux
  - Including BEA, BMC, Computer Associates, Oracle, Peoplesoft, SAP and Veritas

Upcoming Red Hat Sessions

O36	**Technical Overview: Linux 2.6 kernel features and Red Hat Enterprise Linux**				
	Wednesday	08:30 am	-	09:15 am	Salon 3
	Thursday	04:00 pm	-	05:15 pm	Salon 3
O37	**Red Hat Enterprise Linux Security**				
	Thursday	08:30 am	-	09:15 am	Salon 3
	Friday	08:30 am	-	09:15 am	Salon 4
O38	**Migrating from Solaris/Unix to Red Hat Enterprise Linux**				
	Thursday	02:15 pm	-	03:30 pm	Salon 3
	Friday	10:15 am	-	11:30 am	Salon 12
O39	**Linux Integration with Windows Using Samba**				
	Tuesday	04:15 pm	-	05:30 pm	Salon 3