Linux 2.6 Performance in the Corporate Data Center

Open Source Development Labs (OSDL)
Mary Edie Meredith, Data Center Linux TWG Chair

IBM Linux Technology Center (LTC)
Duc Vianney, PhD

Linux World Expo, January 2004

Objectives

- Describe new features resulting from 2.5/2.6 Linux® Kernel development that are likely to improve performance for Data Center applications.
- Understand when application and database servers are likely to benefit from those features.
- Show results of workloads with the 2.4/2.6 Kernel.
- Demonstrate the level of testing that has occurred.

Overview

Perspective

Performance Enhancements 2.6 Kernel

Performance Studies

Stability and Testing Efforts in 2.5/2.6

Summary and References

OSDL

Perspective

IBM

- Discussion is about mainline Linux kernels.

  ---Available from kernel.org

- The focus is on improvements for Data Center Centric workloads on server class machine (4+ CPU) with large memory (4GB+).

- Some improvements may require database code changes for databases to exploit them.

- The improvement depends on your workload.

Overview

Perspective

Performance Enhancements 2.5/2.6 Kernel

Performance Studies

Stability and Testing Efforts for 2.5/2.6

Summary and References

Performance Enhancements 2.5/2.6 Kernel

- **To advance the adoption of Linux in the Data Center, great efforts were made so that the 2.6 kernel would support more memory, cpus, I/O configurations, tasks, and speed than ever before.**
- **Major areas of work that resulted from those efforts:**
  - Virtual Memory
  - I/O improvements
  - Journaling File Systems
  - Task Scheduler
  - Other kernel and scalability efforts
    - Support for large number of tasks
    - Improvements for SMP and NUMA architectures

Virtual Memory

Areas of concern:

When lots of database processes attempted to share large amounts of memory (>=8GB)

Problems with low memory consumption

Performance degradation under memory pressure (swapping)

Features developed to address the concerns:

- Large page support
- Huge TLBfs
- PTE entry placement in high physical memory
- Discontiguous memory support
- RMAPs (Reverse Mapping)

OSDL

Virtual Memory

IBM

Large page support

- Kernel uses PTE's (Page Table Entry) to translate logical addresses to physical addresses.
- Each process has a Page Table with an entry for every page of memory accessed. Other processes that share that page also have a page table entry for it.
- 4 byte entry with 1024 processes would use 4k of low memory to support a 4k sized page (the default).
- A 16GB database buffer uses 16M of low memory per process for PTEs, 1GB for 64 processes --- all of low memory!

Virtual Memory (cont)

Large page support (cont)

- Now can have a PTE point to a very large contiguous block of memory (2M/4M for Pentium® based).
- Improvements:
  - Reduces memory needed for PTEs reducing pressure on low memory
  - Pages are locked into memory
  - Table Look-aside Buffer (TLB) more likely to stay cached (database gets faster access to memory)

Virtual Memory (cont)

HugeTLBfs

- Provides a mechanism for the database to share memory consisting of Large Pages
- Via mmap or shared memory (shmget/shmat)
  - Flag used for shmat/shmget system calls
  - Mount hugetlbfs file system type for mmap calls
  - Sys admin sets aside contiguous memory for this, usually at boot time
  - Kernel configuration parameter to activate support

Virtual Memory (cont)

- **PTE entry placement in high physical memory**
  - Removes low memory (below 1GB) constraint for storing PTEs.
    - DB servers on large systems/processes are less likely to exhaust low memory due to PTEs.
- **Discontiguous memory support**
  - Allows use of all memory even with *holes* in the physical address space on NUMA systems
  - Used for NUMA memory allocation
- **RMAPs (Reverse Mapping)**
  - Previously needed to search all PTEs to find processes referencing a page of memory.
  - Link list of all PTEs referencing it
    - Speeds swapping for memory constrained configurations

I/O - Overview

- **Accessing I/O**
  - Async/direct/raw
- **Handling I/O requests**
  - Locking, elimination of bounce buffers
- **I/O Structure**
  - Block I/O, sector sizes, number of devices
- **Scheduling I/O requests**
  - Scheduler Algorithm, Queues
- **Handling Network I/O**
  - Network Segmentation
  - Network Interrupt handling

I/O -- Direct Accessing

Added Direct I/O for files

- Provides unbuffered I/O for file systems
- Previously only supported for raw devices
  - Releases memory for database use that would otherwise be duplicated in the page cache
  - Narrows the performance gap for choosing file systems over raw
  - raw and O_DIRECT perform comparably (within 2%)

I/O -- Direct Accessing

Added Direct I/O for files (cont)

- The alignment factor for O_DIRECT I/O was reduced from 4096 to 512 byte boundaries.
- O_DIRECT support in 2.6 includes file system types ext2, ext3, xfs, nfs and jfs
- O_DIRECT and raw I/O code was consolidated and re-designed for 2.6
  - Pre-allocation of kiobufs and buffer heads was eliminated
  - No longer breaking the request into smaller-sized chunks (aka Large Block I/O)

I/O - Async Accessing

Async I/O

- Asynchronous I/O for raw devices was added
  - I/O will not block when submitted, allowing many outstanding requests
  - Can greatly improve database I/O throughput
- Async can be combined with direct I/O
- Development on Async I/O for file systems continues.

I/O - Async Accessing

Async I/O (cont)

- AIO System Calls
  - io_setup--- create an aio context capable of receiving the specified number of events
  - io_destroy---destroy an aio_context
  - io_submit--- queue the specified number of iocbs for processing
  - io_cancel--- cancel a previously submitted iocb

I/O - Handling I/O Requests

Locking I/O Request

- Finer grained locking in kernel for I/O requests
  - Linux 2.4 used a single *io_request_lock* spin lock for the entire block device subsystem.
  - Linux 2.6 replaced the *io_request_lock* with more granular locking, which includes a separate lock for each individual device queue.
    - Supports more simultaneous I/O operations
    - Improves overall system I/O throughput on SMP systems with multiple I/O controllers under heavy database load
    - Much higher practical limit of devices per system and per controller

I/O - Handling I/O Requests

Bounce Buffer Avoidance

- In Linux 2.4, a "bounce buffer" is allocated in low memory (below 1GB) when DMA I/O must be performed to or from high memory (above 1 GB)
- In Linux 2.6, device drivers register whether they support high-memory DMA, and bounce buffers can be avoided altogether.
- Improvements:
  - Eliminate copy overhead and memory wastage

I/O - Structure

Device Support

- Device number support for more SCSI disk devices
  - Larger Major and Minor numbers supported
    - 255/255 ---> 2^{12} / 2^{20}
- This increases max possible database size
- Allows more devices for greater I/O response and throughput

I/O - Structure (cont)

- **Large Block I/O**
  - Previously I/Os were broken into 512 byte requests and the results were reassembled.
  - Now requests are made as one large block
    - Reduces overhead for I/Os greater than 512 bytes
    - Big win for databases that typically use 2K and greater.
- **64 bit sector sizes**
  - Can support very large block devices (8 Exa bytes)
    - Greater flexibility in disk configurations with large disk arrays

I/O - Scheduling I/O Requests

Scheduling Algorithms

- Improved I/O scheduling
  - The default scheduler improved to avoid latency problems
    - May improve response time depending on I/O mix
  - Made it possible to offer multiple I/O scheduling options
    - This may allow you to pick the best for your workload

I/O - Scheduling I/O Requests

Scheduling Algorithms (cont)

- Several options are in the baseline and in experimental kernels
  - Deadline (the default for most of 2.5.x)
  - Anticipatory scheduler (AS - the default currently)
  - Noop scheduler
- Deadline is current choice for database workloads
- Scheduler selection via a command line boot option (elevator=deadline)

I/O - Networking

TCP Segmentation off-loading

- Off loads the work of segmenting (packet creates, checksums, enveloping with headers, packet transfer to the NIC).
  - Reduces processor overhead, freeing cycles for database activity due to more efficient use of DMA.

I/O - Networking

- **NAPI (New API)**
  - New polling (epoll) and interrupt handling for network device requests
  - Uses an interrupt approach under light load and polling under heavy
    - Better overall network performance under varying loads.

Journaling FS - Overview

- Review of available file systems, features, performance characteristics
- Extended Attributes
- Access Control Lists

EXT3 and ReiserFS

- **Ext3**
  - Compatible with Ext2
  - Both meta-data & user data journaling
  - Block type journaling
  - 2.4.15, available start of 2.5.x
  - Uses Big Kernel Lock (hurts scalability)
  - Red Hat's default File System at Version 8.0 and later

  Block management:
  - Bitmap based, linear search methods less efficient and *less scalable*

- **ReiserFS**
  - New file layout
  - Balanced trees
  - Good performance with small files
  - 2.4.1, available start of 2.5.x
  - Uses Big Kernel Lock (hurts scalability)
  - SuSE's default File System

  Block management:
  - Bitmap based, linear search methods less efficient and *less scalable*
  - block based

XFS and JFS

- **XFS**
  - Port from IRIX
  - Transaction type journaling
  - External patch available for 2.4.x, added to 2.5.36
  - Provides own locking

  Block management:
  - Use of binary trees improves efficiency and scaling
  - Extent based
  - Allocation Groups

- **JFS**
  - Port from OS/2® Warp Server, code base also used for AIX® JFS2
  - Transaction type journaling
  - 2.4.20, added to 2.5.6
  - Provides own locking

  Block management:
  - Use of binary trees improves efficiency and scaling
  - Extent based
  - Allocation Groups

File System Features and Limits

Features	ReiserFS	Ext3	XFS	JFS
Dynamic inodes	Yes	No	Yes	Yes
Can be /root partition?	Yes	Yes	Yes	Yes
Journal on separate partition	Yes	Yes	Yes	Yes
Online partition re-sizing	Yes	Yes	Yes	Yes
Max. files	4G	4G	4G	4G
Max possible Subdirs/dir	65k	32k	4G	65k
Max. file size	16TB	2TB	16TB	16TB
Max. file system size	16TB	16TB	16TB	16TB

Support for Extended Attributes & Access Control Lists

- Extended Attributes (EA) are arbitrary name/value pairs that are associated with files or directories
  - Maximum EA size is 64K
- **Support for Access Control Lists (ACLs)**
  - Support more fine-grained permissions
  - Store ACLs as Extended Attributes
- **Support EAs and ACLs (2.5.48)**
  - Ext2, Ext3, XFS, JFS
- **Extended Attributes and ACLs for Linux**
  - http://acl.bestbits.at/

Kernel -- Task Scheduler

O(1) scheduler

- New task scheduler for the kernel whose cost stays constant as the number of tasks increases
- Run-queues and locks now on a per CPU basis
  - Improves process and thread scalability
  - Benefits Apps servers and databases on large systems with many connections and processes.

Other Kernel and Scalability Improvements

Preemptible kernel support
NUMA Support
Linux Threads vs NPTL
SYSENTER
New platforms Support
Measurement Tools

Kernel - Preemptible

Preemptible kernel support

- Some kernel routines can now be interrupted
  - Reduces the latency of the kernel, improving overall system performance
  - Targeted to real-time particularly in multi-media applications
  - Selectable as a configuration option

Kernel --NUMA

NUMA Support

- NUMA aware extensions to O(1) scheduler
  - Attempts to run tasks on nodes to optimize performance
    - Increases the likelihood that memory references are local rather than remote for NUMA systems
    - Adds node balancing to existing cpu balancing of activity
    - Work continues in this area for all non-uniform topologies, e.g., HyperThreading.
    - Most 8 way and higher systems are NUMA-like and can benefit.

Linux Threads vs NPTL

- **Linux Threads**
  - **Pros**
    - Fast on UP machines
  - **Cons**
    - Not POSIX compliant
    - Doesn't scale
    - Bogs down on SMP machines
- **NPTL**
  - **Pros**
    - Local thread memory (very fast on UP)
    - Will be POSIX compliant
    - Tightly integrated with kernel
    - Likely to become next default
  - **Cons**
    - Immature
    - Incompatible with most Linux Threads apps
    - 1:1 threading model
    - Kernel resource intensive

Kernel -- SYSENTER

- **SYSENTER**
  - Faster System Calls via SYSENTER extension if supported in hardware
    - Faster change from user mode to kernel mode, reducing the overhead of a system call
    - Requires updated glibc and gcc
    - Improvement is for Pentium 4

Kernel -- Ports

- **New (or improved) platforms**
  - 64 Bit PowerPC®
  - X86-64 AMD Opteron(TM)
    - 512MB per process limitation removed
    - 32bit support improved
  - User Mode Linux (UML) -- a useability feature.

Kernel - Measurement Tools

- **Measurement Tools**
  - O-profile System wide performance profiler
    - Collect user and kernel activity (readprofile is limited to kernel only)

Overview

Perspective

Performance Enhancements 2.5/2.6 Kernel

Performance Studies

Stability and Testing Efforts in 2.5/2.6

Summary and References

Performance Studies
Workloads Used for Comparisons

Selected Workloads	Test Suite	Test Location
OLTP DB Server	DBT-2	OSDL
Microtests (AIM7/9)	ReAIM	OSDL
Web Server	SPECweb99	IBM® LTC
Application Server	SPECjAppServer	IBM LTC
Java Dev, Hyperthreading	SPECjbb2000	IBM LTC
Data Warehousing, IO Sched	Decision Support	IBM LTC

Performance Studies
Workloads Used for Comparisons

Selected Workloads	Test Suite	Test Location
OLTP DB Server	DBT-2	OSDL
Microtests (AIM7/9)	ReAIM	OSDL
Web Server	SPECweb99	IBM LTC
Application Server	SPECjAppServer	IBM LTC
Java Dev, Hyperthreading	SPECjbb2000	IBM LTC
Data Warehousing, IO Sched	Decision Support	IBM LTC

OSDL Database Test (DBT) Suite

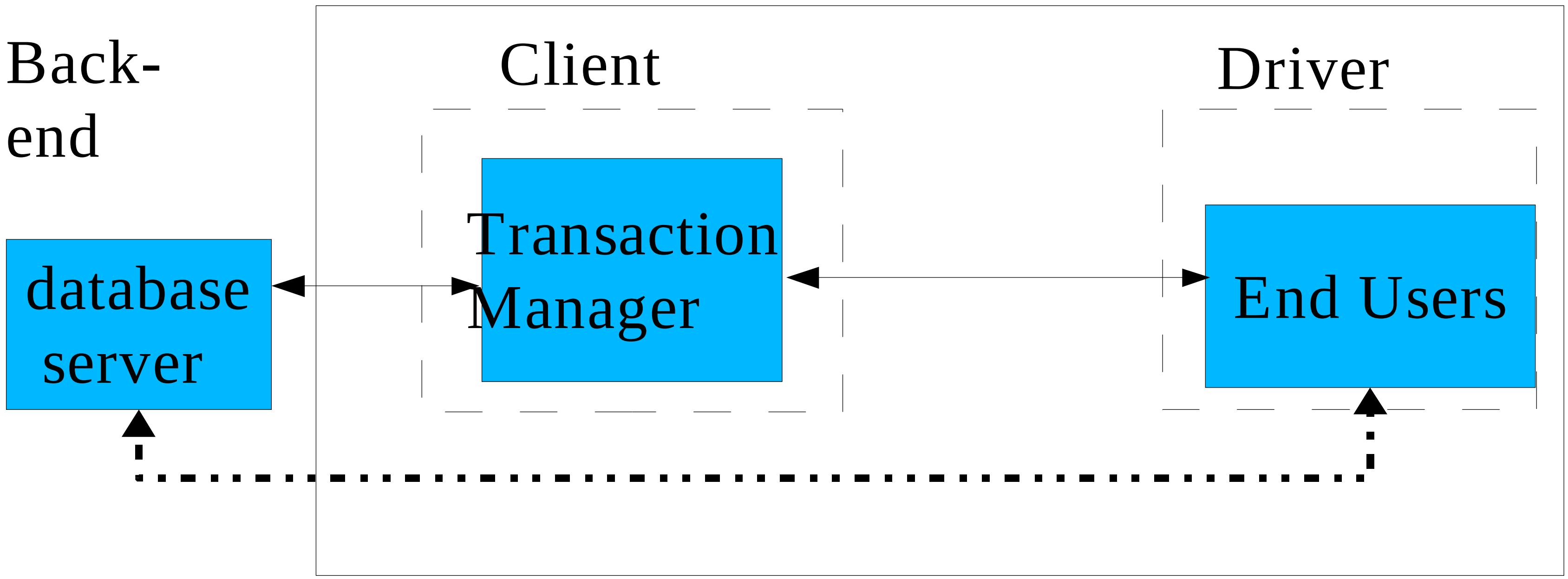
Workloads based on Transaction Processing Council (TPC) Benchmarks Specifications

- Open Source Kit, currently supports:
  - SAP-DB
  - PostgreSQL
- Appropriate for comparing kernels.
- NOT appropriate for comparing hardware or RDBMS software.
- Database Test 2 (DBT2) is a fair use implementation of TPC-C
  - Activities of a wholesale parts supplier

OLTP Database Server
Database Test 2 (DBT2)

- DBT2 Workload Components

  Focused on demands for a database server.



Workload variables:

 # database connections (drivers or TM client)

 # warehouses (determines database size)

 # warehouses touched by drivers (run set size)

 # transaction mix (5 transactions)

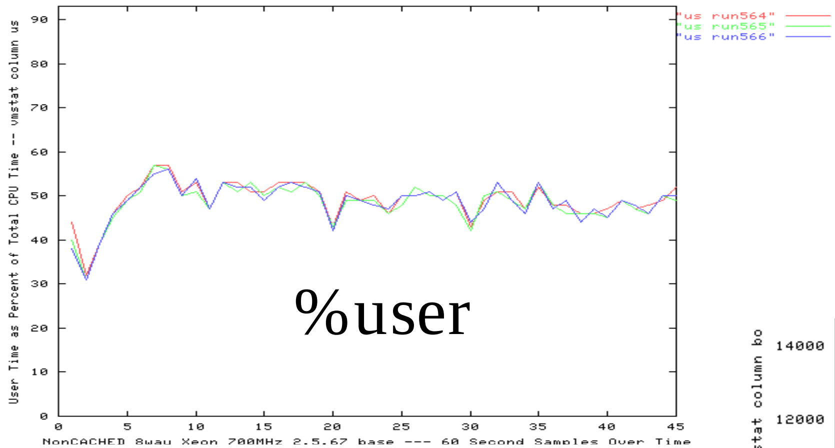
#keying time and think time

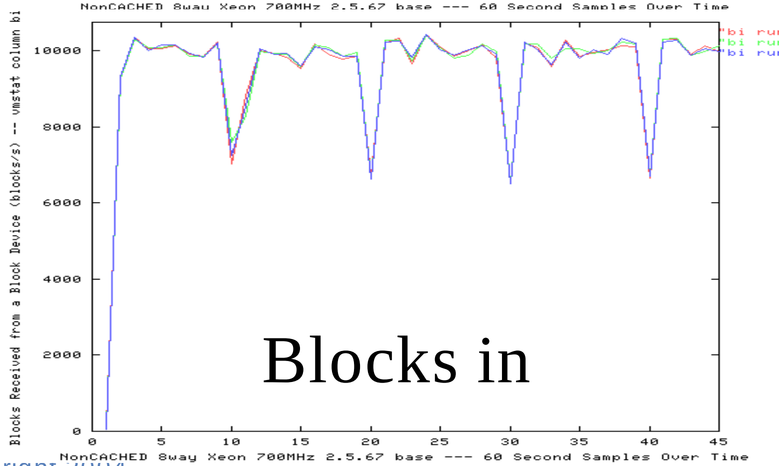
DBT2 8 way Test Configuration

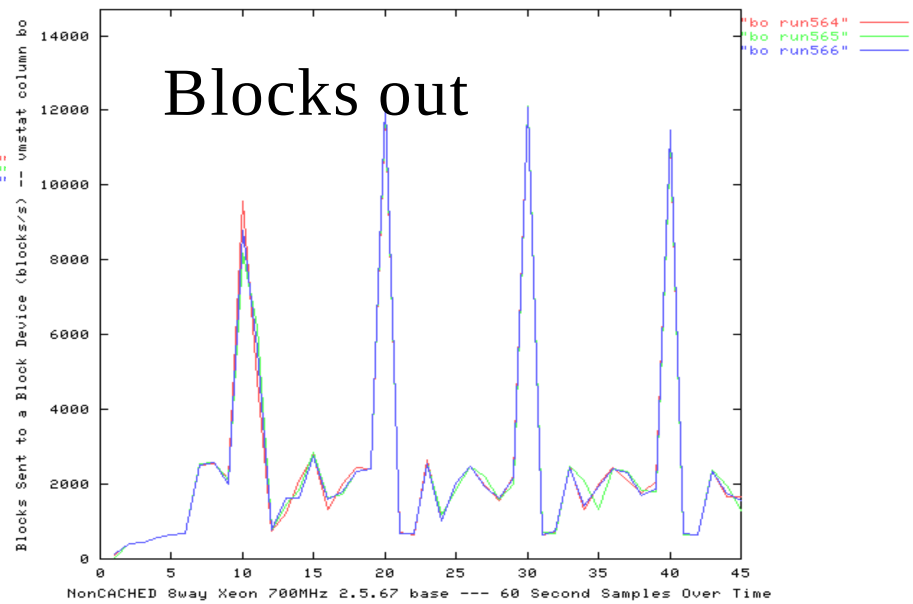
- **DBT2 Test choices**
  - Warehouses 100 with 11GB database size
  - Warehouses touched by drivers (run set size)
  - Transaction mix (5 transactions)
  - Keying time and think time - zero
  - Driver only emulation -- driver and database on the same system.
- **Two variants**
  - Cached - 8 drivers with cached working set for system memory
  - Non-cached - 16 drivers touching 96 warehouses
- **Database System Equipment**
  - 8 CPU Pentium III 1Mbyte cache, 4GB memory
  - 12 - 10k rpm 72GB drives configured as raw devices
- **SAP-DB 7.3.0.25**
- **User space RH7.3/RH9.0 for the database**

DBT2 Non-cached Workload Characteristics

- Non-cached case: Random read, writes, 8k
- Synchronous writes to log file
- No swapping, high active memory

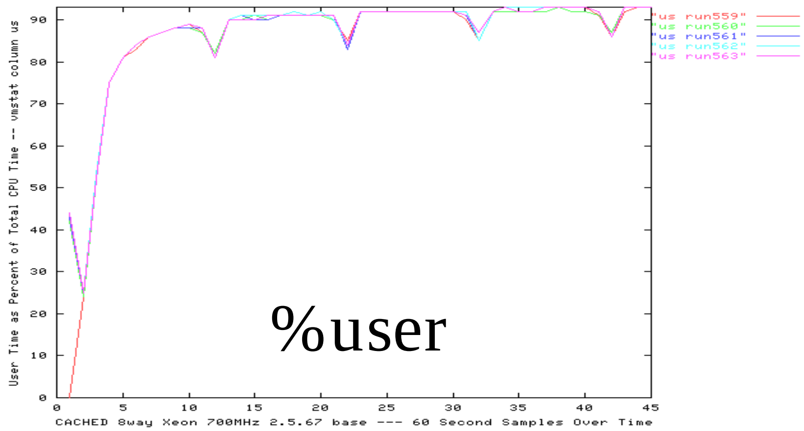


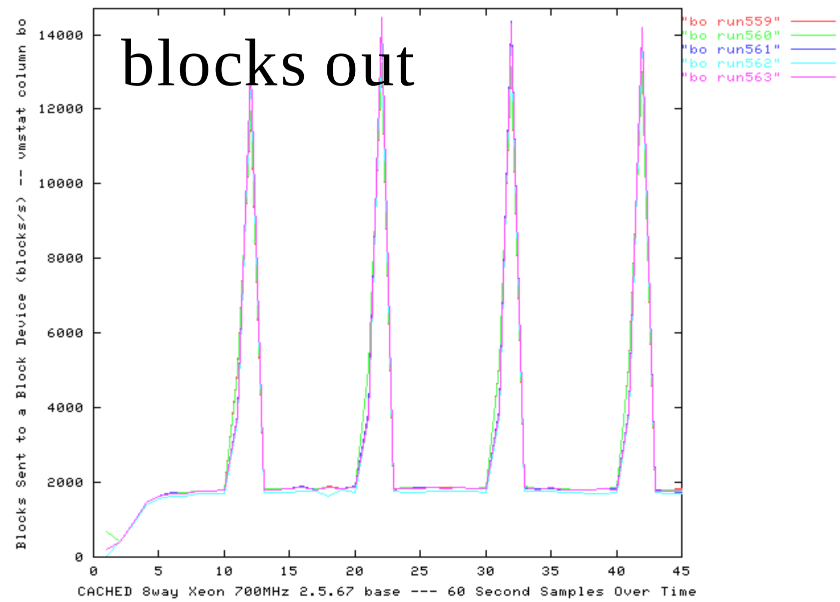


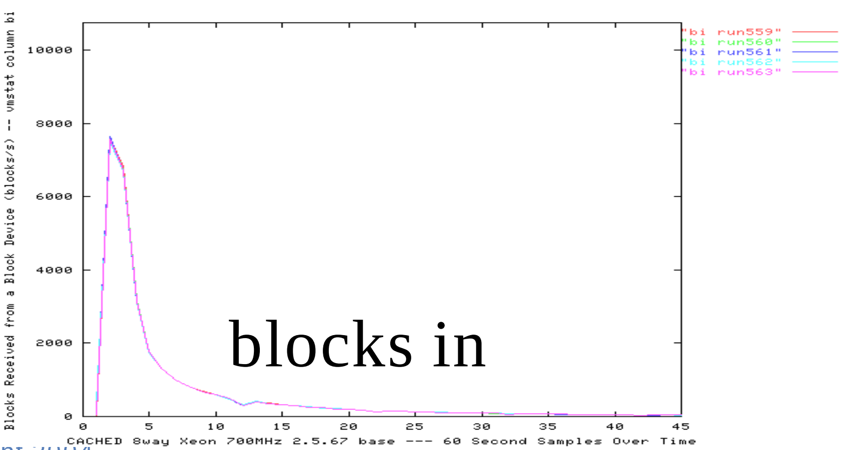


DBT2 Cached Workload Characteristics

- Cached case heavy CPU activity, non-cached I/O bound
- Heavy synchronous writes to log file
- No swapping, high active memory







OSDL

DBT2 8-way Test Results

IBM

Metric is NOTPM, New Order Transactions per minute, (bigger is better)

	cached 2.4.21rc1	cached 2.6.0.test11	non-cached 2.4.21rc1	non-cached 2.6.0.test11
Average NOTPM	4533.8	4925.4	1413.33	1696
% Improvement		8.64		20.00

Improvement over 2.4 is greater with increased I/O workload

OSDL ReAIM Tests

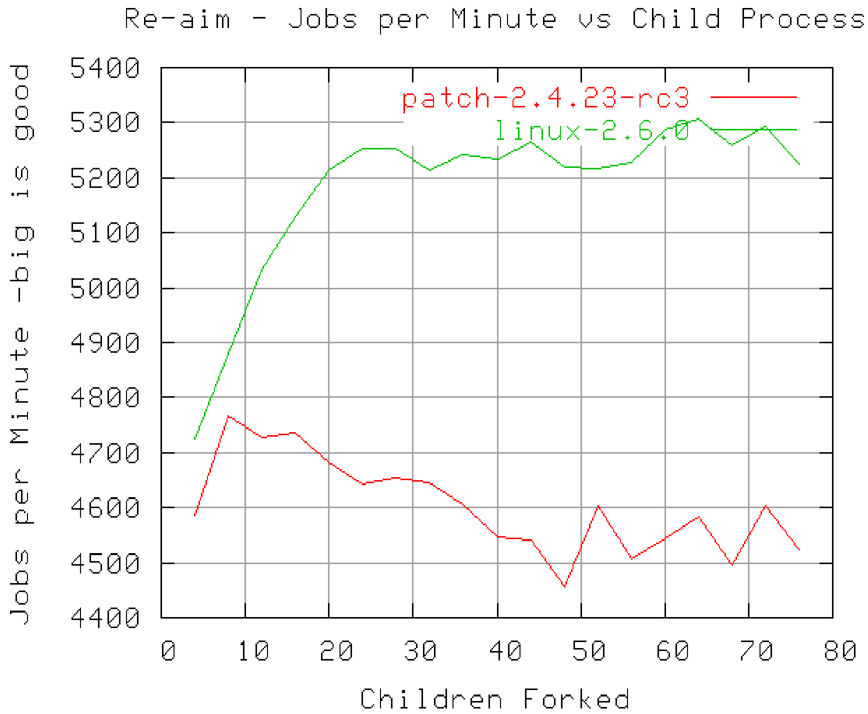
- OSDL ReAIM run at OSDL Labs
  - Combination of AIM7 and AIM9 micro-test suites
  - Added features
  - Used the Database Mix
- Test Configuration 4-way and 8-way STP (OSDL Scalable Test Platform) systems
  - 4 /8 CPU 700MHz Pentium III 1M cache
  - 4GB/8GB memory
  - Qlogics Fibre Channel disk controller ISP2000
  - 2 SCSI drives 18GB IBM-PSG model ST318304FC

Intended to test scheduler with micros tests

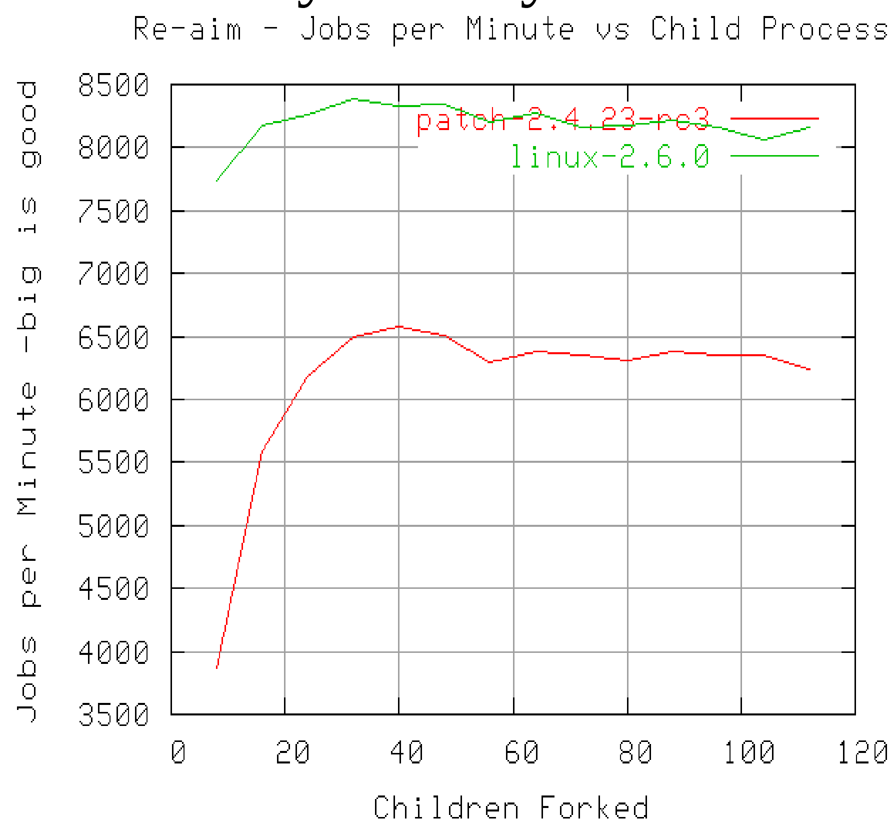
ReAIM Results

2.4.23rc2 versus 2.6.0

4 way STP system



8 way STP system



2.6 peaks at a higher throughput and holds the throughput number higher with more processes.

Performance Studies
Workloads Used for Comparisons

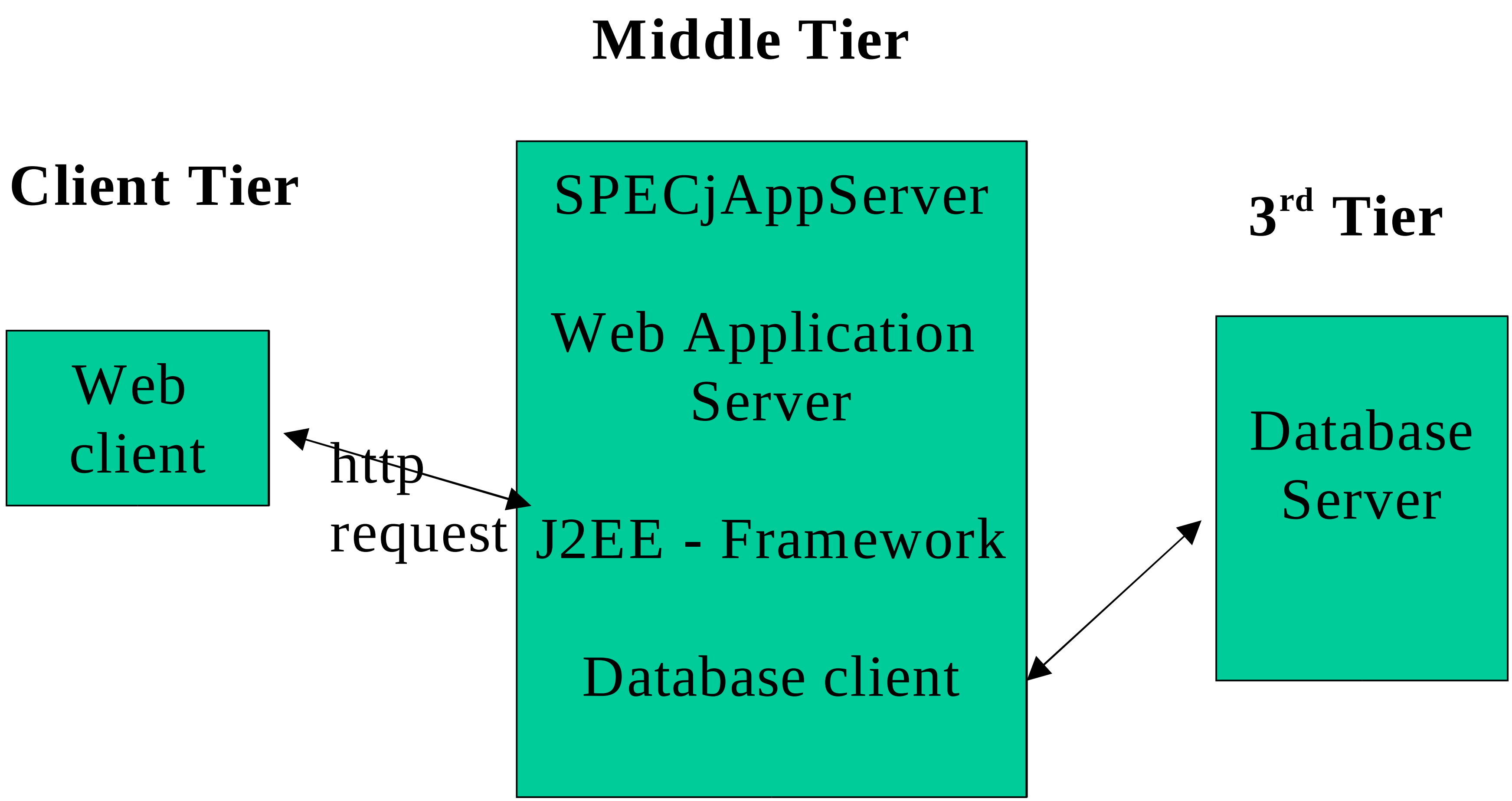
Selected Workloads	Test Suite	Test Location
OLTP DB Server	DBT-2	OSDL
Microtests (AIM7/9)	ReAIM	OSDL
Web Server	SPECweb99	IBM LTC
Application Server	SPECjAppServer	IBM LTC
Java Dev, Hyperthreading	SPECjbb2000	IBM LTC
Data Warehousing, IO Sched	Decision Support	IBM LTC

SPECweb99 on 4-, 8-way

- **Web Server benchmark**
- **Hardware and Software:**
  - 8-way, 900 MHz, 2MB L2, 28 GB RAM, (4) e1000, Apache 2.0.43+mod_specweb
- **2.6.0-test2 vs. 2.4.21:**
  - 4-way: 26%
  - 8-way: 35%

Note: SPECweb99 is a trademark of the Standard Performance Evaluation Corp. (SPEC). The SPECweb99 results or findings in this publication have not been reviewed or approved by SPEC, therefore no comparison or performance inference can be made against any published SPEC results.

Web App Server Workload Overview:SPECjAppServer



SPECjAppServer2002 3-Tier configuration

SPECjAppServer2002 Test Configuration

- **Emulates a manufacturing, supply chain, and order/inventory system**
- **Web Application Server**
  - 4- and 8-way 2.0 GHz, WebSphere® Application Server 5.0.2/JDK 1.3.1
- **Database Server**
  - 4-way 700 MHz, IBM DB2® 8.1 FP3
- **Client System**
  - 2-way 1.0 GHz
- **Configuration Options**
  - Connections between web application server and database server (100)
  - Number of threads in the web application server (50)

SPECjAppServer2002 on 4-, 8-way

- **2.6.0-test7 vs. 2.4.21:**
  - 4-way: 7%
  - 8-way: 14%

Note: SPECjAppServer2002 is a trademark of the Standard Performance Evaluation Corp. (SPEC). The SPECjAppServer2002 results or findings in this publication have not been reviewed or approved by SPEC, therefore no comparison or performance inference can be made against any published SPEC results. The official web site is located at

http://www.spec.org/osg/jAppServer2002.

SPECjbb2000 on 4-, 8-way

- **Emulates warehousing, order entry system**
- **Hardware and Software:**
  - 4-, 8-way 1.5 GHz, 512KB L3, 8 GB RAM, IBM JVM 1.4.1
- **2.6.0-test2 vs. 2.4.21:**
  - 4-way: 6%
  - 8-way: 8%

Note: SPECjbb2000 is a trademark of the Standard Performance Evaluation Corp. (SPEC). The SPECjbb2000 results or findings in this publication have not been reviewed or approved by SPEC, therefore no comparison or performance inference can be made against any published SPEC results.

Hyperthreading- SPECjbb2000

- **Hardware and Software:**
  - 4-way, 1.5 GHz, 512KB L3, 8 GB RAM, IBM JVM 1.4.1

- **Hyperthreading vs. Non-hyperthreading:**
  - 2.5.69: 14%

Note: SPECjbb2000 is a trademark of the Standard Performance Evaluation Corp. (SPEC). The SPECjbb2000 results or findings in this publication have not been reviewed or approved by SPEC, therefore no comparison or performance inference can be made against any published SPEC results.

Decision Support on 8-way

- **Emulates a Data Warehouse workload**
- **Hardware and Software:**
  - 8-way 2.0 GHz system, 2MB L3 cache, 16 GB RAM, 4 QLogic 2300 controllers, 2 FAStT900 storage devices, 112 disks, HT enabled, DB2 v8.1+SP4 early release candidate, large pages, qla2xxx v8.3, 100 GB database.
- **2.6.0-test5 vs. 2.4.21:**
  - 8-way: 10% faster

I/O Scheduler on 8-way

- **Hardware and Software:**
  - 8-way 2.0 GHz system running a 100 GB database, 2MB L3 cache, 16 GB RAM, 4 QLogic 2300 controllers, 2 FAStT900 storage devices, 112 disks, HT enabled, Decision Support workload.

- **2.6.0-test5: Anticipatory Scheduler vs. Deadline Scheduler:**
  - light load: same performance
  - heavy load: AS is ---27% against Deadline

Results Summary

Workload	Test Suite	Results Relative to 2.4	
		% improve	% improve
		4 way	8way
OLTP DB Server	DBT-2 non-cached	n/a	8.64
OLTP DB Server	DBT-2 cached	n/a	20.00
Microtests (AIM7/9)	ReAIM	9.75	27.78
Web Server	SPECweb99	26	35
Application Server	SPECjAppServer	7	14
Java Development	SPECjbb2000	6	8
Hyperthreading	SPECjbb2000	14*	n/a
Data Warehousing	Decision Support	n/a	10

* hyperthreading versus non-hyperthreading on the same kernel

Overview

Perspective

Performance Enhancements 2.5/2.6 Kernel

Performance Studies

Stability and Testing Efforts in 2.5./2.6

Summary and References

Linux 2.5/2.6 Stability

OSDL Production System uptime reports from 12/30/03

Server Function:	www.osdl.org	Internal Network Server Master	Internal Network Server Slave	PLM Compile Machines
Linux Kernel:	linux-2.5.66 (PLM Id)	linux-2.5.66 (PLM Id)	linux-2.5.66 (PLM Id)	linux-2.5.66 (PLM Id)
Applications:	Apache 2 Bind Sendmail SquirrelMail Mailman Bugzilla	Bind DHCP LPRng	Bind DHCP LPRng	
System Uptime:	22 days - 5:13	158 days - 10:39	221 days - 20:14	92 days - 19:59
Load Average:	1.76, 0.81, 0.58	0.56, 0.14, 0.04	0.00, 0.00, 0.00	0.00, 0.00, 0.00

Tables last updated Tue Dec 30 17:10:08 2003

For updates and more about Linux Stability efforts see :
http://www.osdl.org/projects/26lnxstblztn/results/

Linux 2.5/2.6 Regular Testing

- Test Suites are run against 2.5/2.6 kernels on a regular basis:
  - Linux Test Project Suite: variety of functional tests to validate system calls (http://ltp.sourceforge.net)
  - Open POSIX test suite: functional test suite to validate POSIX 2001 standards. (http://sourceforge.net/projects/posixtest/)
  - Scalable Test Platform: tests kernel builds and good suite of stress and performance tests (http://www.osdl.org/stp/)

Linux 2.5/2.6 Performance Regression Testing by LTC

- Performance regression against 2.5/2.6 kernels are run on a regular basis:

  - Microbenchmarks: dbench, kernbench, lmbench, rawiobench, tbench, tiobench
  - Application benchmarks: SPECjbb2000, SPECsdet, VolanoMark
  - Results URL: **http://ltcperf.ncsa.uiuc.edu/data/**

Summary

Linux 2.6 integrates many performance enhancements for data center workloads into a new kernel base that :

- Removes many size and scaling barriers
- Has proven performance improvements on data center workloads

Linux 2.6 received an unprecedented variety and level of testing throughout its development cycle.

Opportunity for the community to move quickly to the 2.6 kernel.

Acknowledgements

OSDL:

Cliff White, Mark Wong, Dave Olien

IBM LTC Performance:

Mala Anand, Mark Peloquin, Steve Pratt, Mike Skelton, Mike Sullivan, Andrew Theurer, Troy Wilson, and Peter Wong.

References

developer.osdl.org/maryedie/LWE_NYC04_Links.html

links to more info

Linux Documentation references

link to the presentation

maryedie@osdl.org

dvianney@us.ibm.com

Final Words Regarding 2.5/2.6 Development

Linux 2.6 started with 2.4 as its code base and includes many features that benefit database workloads.

Selected back ports can improve 2.4 performance

A back port ultimately lacks integration with the architectural changes, e.g.:

- Sysfs for common topology interface for device drivers
- Threading improvements (Native POSIX Threading Library, NPTL)
- USB device support improvements

Opportunity for the community to move to the 2.6 kernel, reducing the number of patches maintained by distros, letting them focus *above the kernel.*

Legal Statement