Linux Kernel Performance Measurement and Evaluation

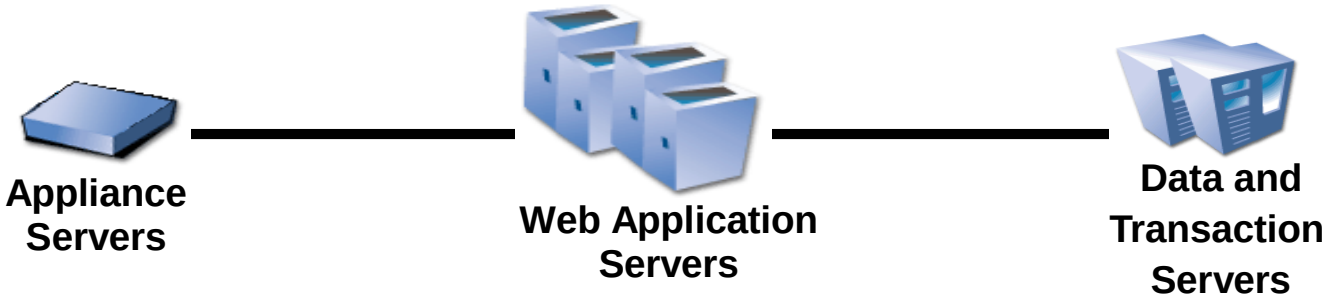
Duc Vianney, Sandra Baylor, Bill Hartner
IBM Linux Technology Center
LinuxWorld / San Francisco
14 August 2002

IBM
Linux Technology Center

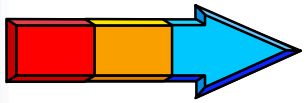
Outline

- Enterprise Linux® Requirements
- Linux Scalability Work Synopsis
- LTC Kernel Performance Team
- Linux Kernel Focus
- Benchmark Activities and Workloads
- Summary of Activities
- LTC Kernel Performance Team Contacts
- VolanoMark Scalability Work Results

Enterprise Linux Requirements



- RAS Trend →
  - reliability, availability, servicability
- Interoperability
- Scalability
  - horizontally scalable (SMPs)
  - vertically scalable (clustering)



Team focuses on "**horizontal scalability**"

Linux Scalability Work Synopsis

- Ultimate goal is to increase overall system and application performance:
  - driven by benchmark analysis and/or customer workloads
- Three major categories need to be addressed
  - resource scalability
  - SMP scalability
  - absolute performance
- Ensure the full utilization of resources (cpu, memory, devices)
- In a perfect world: Linear Scalability

Linux Scalability Community

- **Most scalability work at IBM consists of Open Source projects**
- **Available under:**
  - http://lse.sourceforge.net
  - http://lbs.sourceforge.net
- **Active participants in this effort are:**
  - IBM, SGI, HP, Intel, Hitachi, NEC, SUSE
  - Many individuals out there
- **Strong interactions among participants**
  - to agree on and do the right thing
  - tremendous sharing of code / results / tools
  - regular conference calls and meetings

LTC Kernel Performance Team

- **Mission**
  - To make Linux better by improving Linux kernel performance, with special emphasis on SMP scalability.
- **Methodology**
  - Measure, analyze and improve the performance and scalability of the Linux kernel
  - Focus on platform-independent issues
  - Benchmarks that provide coverage for data center, carrier space and web server workloads
  - Migration to newer kernels will occur as needed
- **Plan Assumptions**
  - Work items may change based on IBM strategy and acceptance from the open source community
  - Work items may change as measurement results unfold and/or hardware requirements increase
  - Baseline measurements currently on Linux 2.4 and 2.5 kernel.org

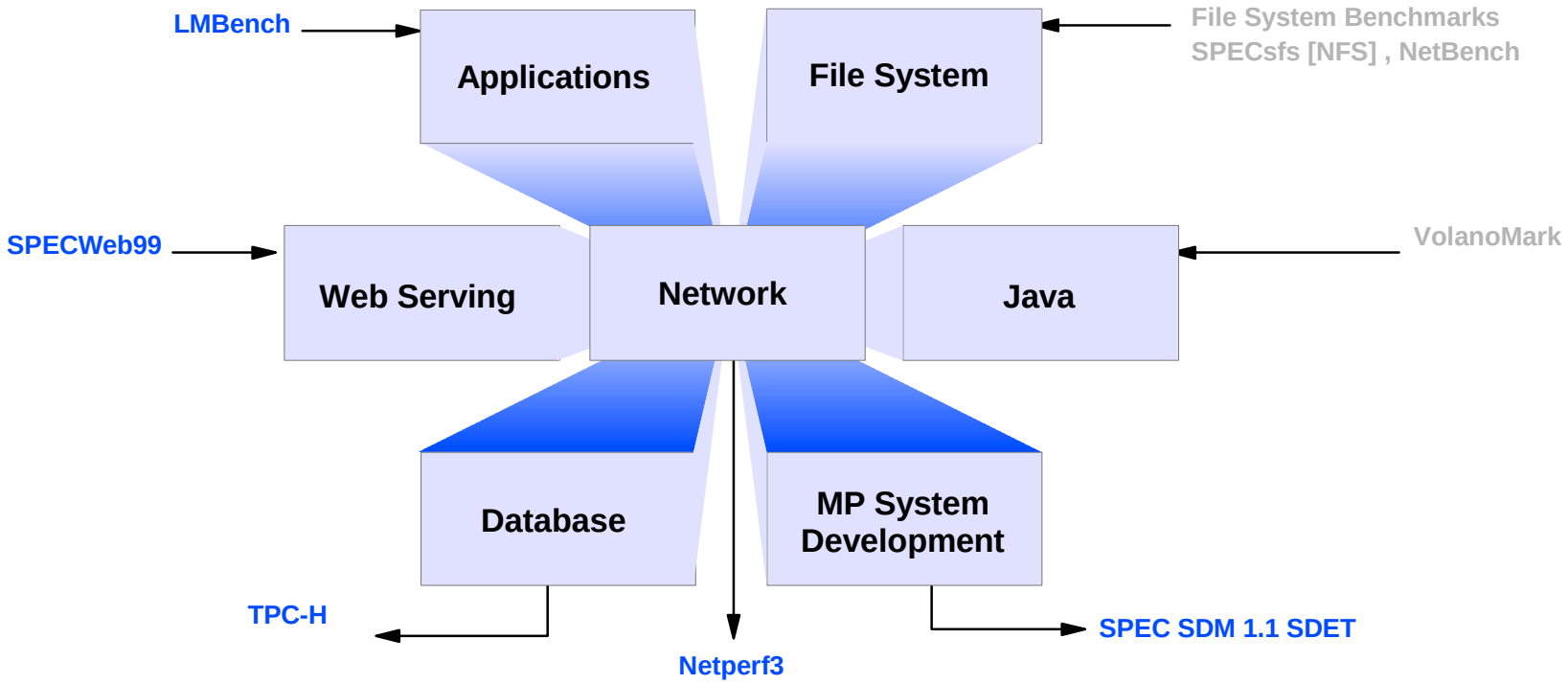
Linux Kernel Focus

- **File System**
  - Local file systems: ext2, ext3, jfs, reiserfs
  - Network file systems: nfs, smb and virtual: vfs
- **Base Kernel**
  - Scheduler
  - Memory Management (large memory support, page cache)
- **Buffer Cache**
- **Peripheral Manager**
  - Block device interface
- **Network Manager**
  - 100 Mbps and 1000 Mbps Ethernet
  - TCP/IP
- **Machine Interface**
  - Interrupt manager (APIC, soft IRQ/bottom half)

High Priority "Enterprise" Workloads

- **Web Serving**
  - Where Linux is traditionally strong
  - Web server; infrastructure servers such as DNS, mail, file/print, etc.
  - Typically 4-way SMP, horizontal scaling
  - Improvement still required for larger SMP, security, Web App serving, Java™,...
- **Backend DB (addresses many other enterprise workload reqs)**
  - DB2®/Oracle/Sybase
  - Typically 8-way SMP or higher, vertically scaled
  - Transaction, backend DB serving - as opposed to high-end decision support (typically requiring horizontally scaled cluster)
- **Telco Carrier Network/Network Infrastructure**
  - Typically 4-way SMP systems deployed in the core Telco network (e.g., softswitches, wireless base station controllers, etc.)
  - HA middleware dependencies (IP spraying, replicated in-mem DBs, DB fast failover...)

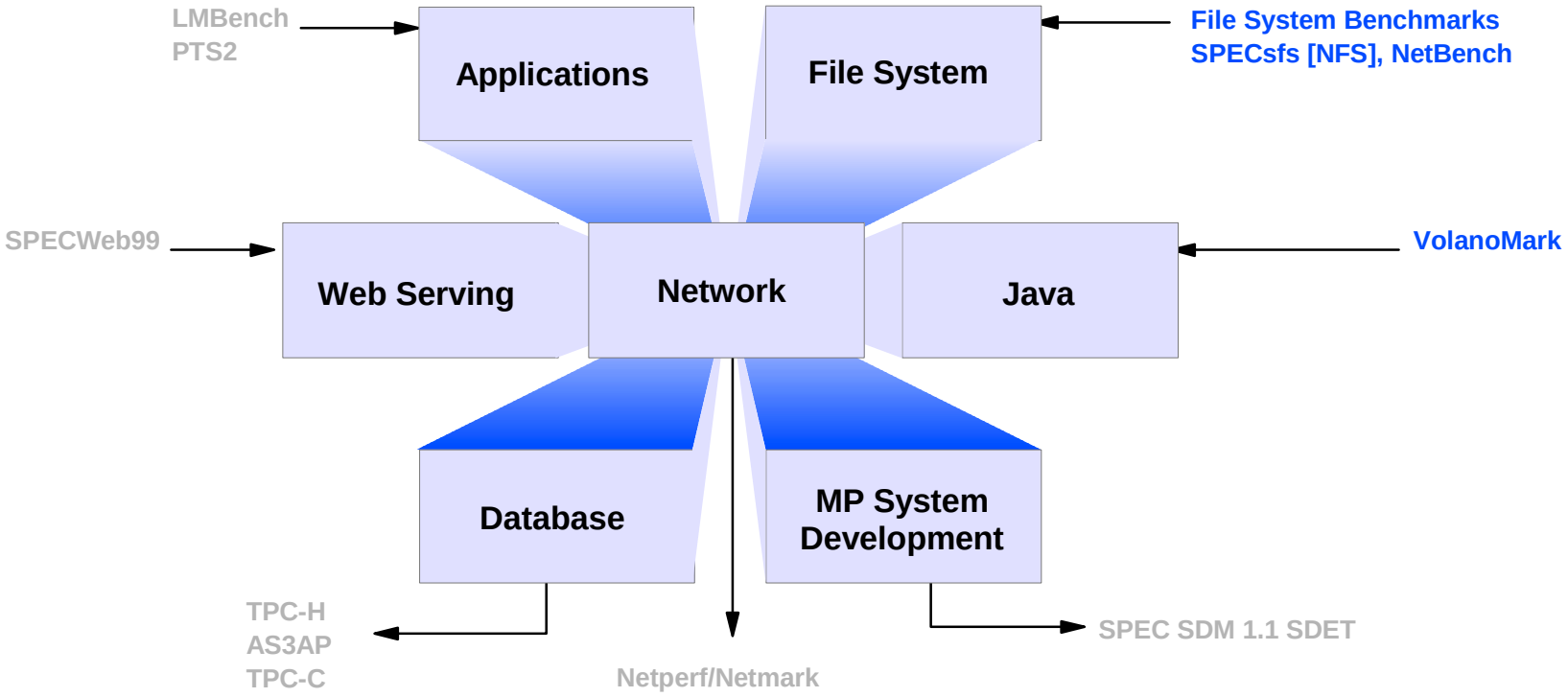
Benchmarks and Workloads



- **LMBench [atomic API test]**
  - Open Source benchmark
  - Linux APIs
- **SPECWeb99 [web serving]**
  - SPEC Industry standard benchmark
  - TCP/IP, network device drivers, memory management, file system
- **TPC-H**
  - industry standard benchmark
  - file system
- **Netperf**
  - open source
  - TCP/IP, network device drivers, memory management
- **SPEC SDM1.1 SDET [multiprocessing system development]**
  - deprecated SPEC benchmark
  - file system, scheduler, memory management

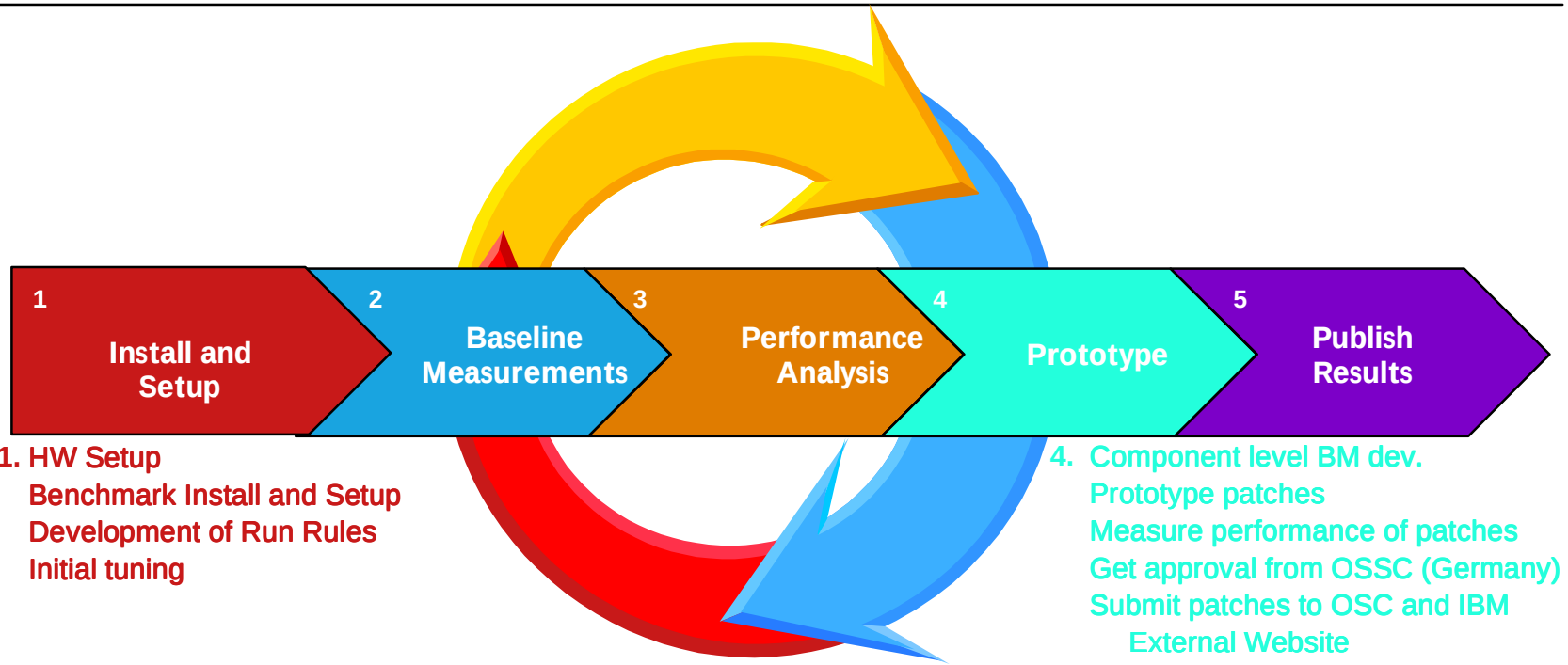
IBM
Linux Technology Center

Benchmarks and Workloads



- **VolanoMark [Java]**
  - industry standard benchmark
  - Java, scheduler, TCP/IP, network device drivers
- **File System Benchmarks**
  - Open Source benchmarks (e.g., dbench, bonnie, iozone, postmark)
  - file system - virtual file system, buffer cache, page cache, block device interface, memory management
- **SPECsfs [NFS file servicing]**
  - SPEC Industry standard benchmark
  - NFS, file system, TCP/IP, network device drivers, memory management
- **NetBench 7.0 w/Samba 2.0.7**
  - Ziff Davis benchmark
  - TCP/IP, network device drivers, memory management, file system

Benchmark Activities



1. HW Setup
   - Benchmark Install and Setup
   - Development of Run Rules
   - Initial tuning

2. Baseline performance/scalability measurements
   - Initial publication of benchmark results to OSC
   - Announcement of BM to OSC
   - Collaboration with OSC

3. Collection of performance analysis data
   - System and benchmark tuning (from analysis data)
   - Performance/scalability re-measurements
   - Identification of potential performance bottlenecks
   - Development of kernel component analysis tools
   - Detailed analysis of potential performance bottlenecks
   - Collaboration with OSC

4. Component level BM dev.
   - Prototype patches
   - Measure performance of patches
   - Get approval from OSSC (Germany)
   - Submit patches to OSC and IBM External Website
   - Collaboration with OSC

5. Publication of papers etc.

Hardware and Software Tuning

- Before any measurements are made, tune HW and SW configurations prior to analysis of performance and scalability
- **Tuning**
  - An iterative cycle of tuning and measuring
  - Involves measuring components of the system
  - CPU utilization, memory usage, etc.
  - Involves possibly adjusting system hardware parameters, system resource parameters, and middleware parameters
  - One of the first steps in performance analysis
- **Performance and scalability analysis**
  - Understand benchmark and workload tested
  - Initial analysis conducted against tuned system
  - Requires a set of performance tools

Performance Tools

- /proc file system - meminfo, slabinfo, interrupts, network stats, io stats, etc.
- profile and readprofile
- SGI's lockmeter - SMP lock analysis
- SGI's kernel profiler (kernprof) - time based profiling, performance counter based profiling, annotated call graph (ACG) of kernel space only
- Ad hoc performance tools are developed to further understand a specific aspect of the system. Examples are:
  - sstat - collects scheduler statistics
  - schedret - determines which kernel functions are blocking for investigation of idle time
  - acgparse - post-processes kernprof ACG
  - copy in/out instrumentation - determines alignment of buffers, size of copy and CPU utilization of copy in/out algorithm

Summary of Activities

- **Linux is regarded as a stable, highly-reliable operating system for web servers -->** low-end to mid range systems
- **More work is needed for Linux to be ready for enterprise markets**
- **LTC Linux Kernel Performance team focuses on addressing issues for enterprise markets**
  - 8-way SMP scalability and beyond
  - Web server, carrier space, database and other workloads
  - Strategy includes the measurement, analysis, and improvements, through kernel patches, to the Linux kernel, focusing on architecture-independent issues
  - Incorporated several optimizations and patches that improve performance of our benchmarks
    - processor and IRQ affinity
    - bounce buffer patch to IPS RAID driver
  - Making great progress towards addressing issues, improving the performance and scalability of Linux so that it is ready for enterprise markets

Volanomark Benchmark Overview

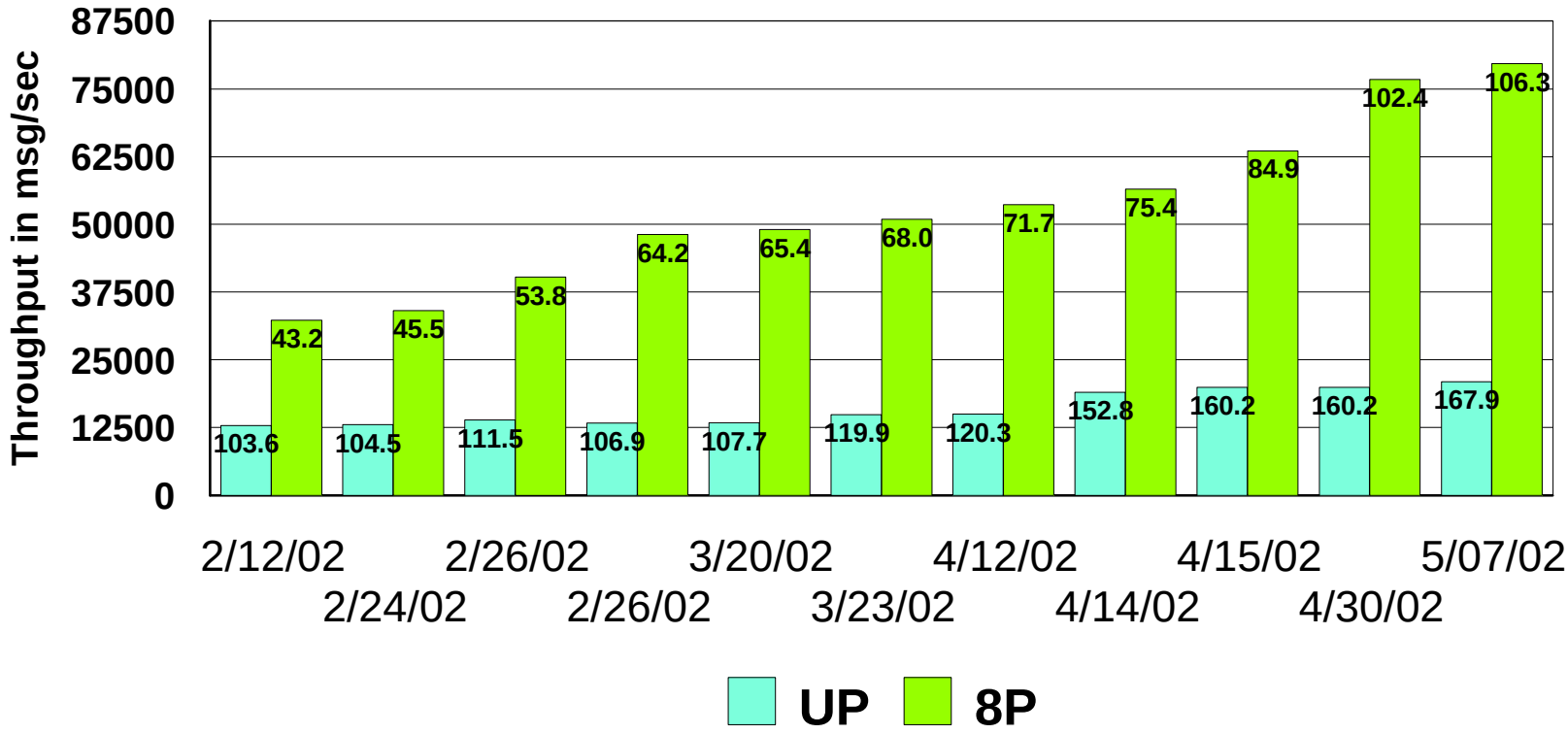
- **Volanomark is a benchmark of a chatroom server**
  - Developed by Volano, LLC
  - Java TCP messaging benchmark (chat room server)
  - Large number of TCP connections and Java threads
  - The benchmark simulates the users in a number of chat rooms.
  - Throughput in messages / second
  - http://www.volano.com
- **Benchmark environment**
  - Hardware:
    - Netfinity® 8500R server, 8x700 MHz, 1 MB L2, 4 GB RAM
  - Software:
    - IBM® JRE 1.3.x
    - Red Hat 7.1 Linux
    - Kernel.org version 2.4.x and 2.5.x

VolanoMark Scalability

VolanoMark Loopback

UP Target: 12,500 msg/sec 8P Target: 75,000 msg/sec

Data Preliminary



Netfinity 8500R, 8-Way 700 MHz Pentium(TM) III with 2MB L2 Cache, 4 GB RAM
Red Hat 7.1, Linux Kernel 2.4.17 + Patches + Tuning
Numbers inside the bars represent % of target achieved

IBM
Linux Technology Center

VolanoMark Performance Issues

- **Signals**
  - JVM using Linux threads make more system calls for signal handling
- **TCP/IP**
  - loopback code path inefficient
  - send and receive thread executed on different cpus
- **Scheduler**
  - runq length
  - Load balancing
  - Priority Preemption

8-way VolanoMark Performance

- **Changes Made to Reach the Most Recent Milestone**
  - Added Priority Preemption patch
  - Added TCP/IP soft affinity patch
  - Tuned TCP/IP (timestamps,softack,hot_list_length)
  - Set the Loopback MTU to 512
  - Tuned Client messages
  - O(1) scheduler and scalable counters
  - JVM 1.3.1
- **Next Steps Towards Improving Performance**
  - NGPT (Next Generation Pthreads): faster locking, and M:N threading will reduce number of kernel thread
- **Investigate TCP/IP optimizations**
  - send/recv code path
  - cache line efficiency
  - loopback driver code path

LTC Kernel Performance Team Contacts

- **Tech Lead:** Bill Hartner - bhartner@us.ibm.com
- **Datacenter/Scalable:** TPC-H (Peter Wong - wpeter@us.ibm.com)
- **Telco Carrier Space:** Netperf3 (Mala Anand - manand@us.ibm.com) and VolanoMark (Partha Narayanan - partha@us.ibm.com)
- **Web Serving:** SPECweb99 (Troy Wilson - wilsont@us.ibm.com)
- **File Serving:** SPECsfs and Netbench (Andrew Theurer - atheurer@us.ibm.com)
- **Realtime, Filesystem and API Benchmarks:** LMBench, IOzone, etc. (Duc Vianney - dvianney@us.ibm.com)
- **I/O Benchmarks:** block I/O, async I/O (Helen Pang - hpang@us.ibm.com)
- **Other Activities:** NUMA (Theurer/Wong), IA64 (Vianney), SPECjAppServer (Ruth Forester - rsf@us.ibm.com)
- http://oss.software.ibm.com/developerworks/opensource/linuxperf
- http://oss.software.ibm.com/developerworks/projects/linuxperf

Legal Statement

This work represents the views of the authors and does not necessarily reflect the views of IBM Corporation.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries: IBM, IBM (logo), Netfinity.

Linux is a registered trademark of Linus Torvalds.

Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

Pentium is a trademark or registered trademark of Intel Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.