Emerging Linux Technologies

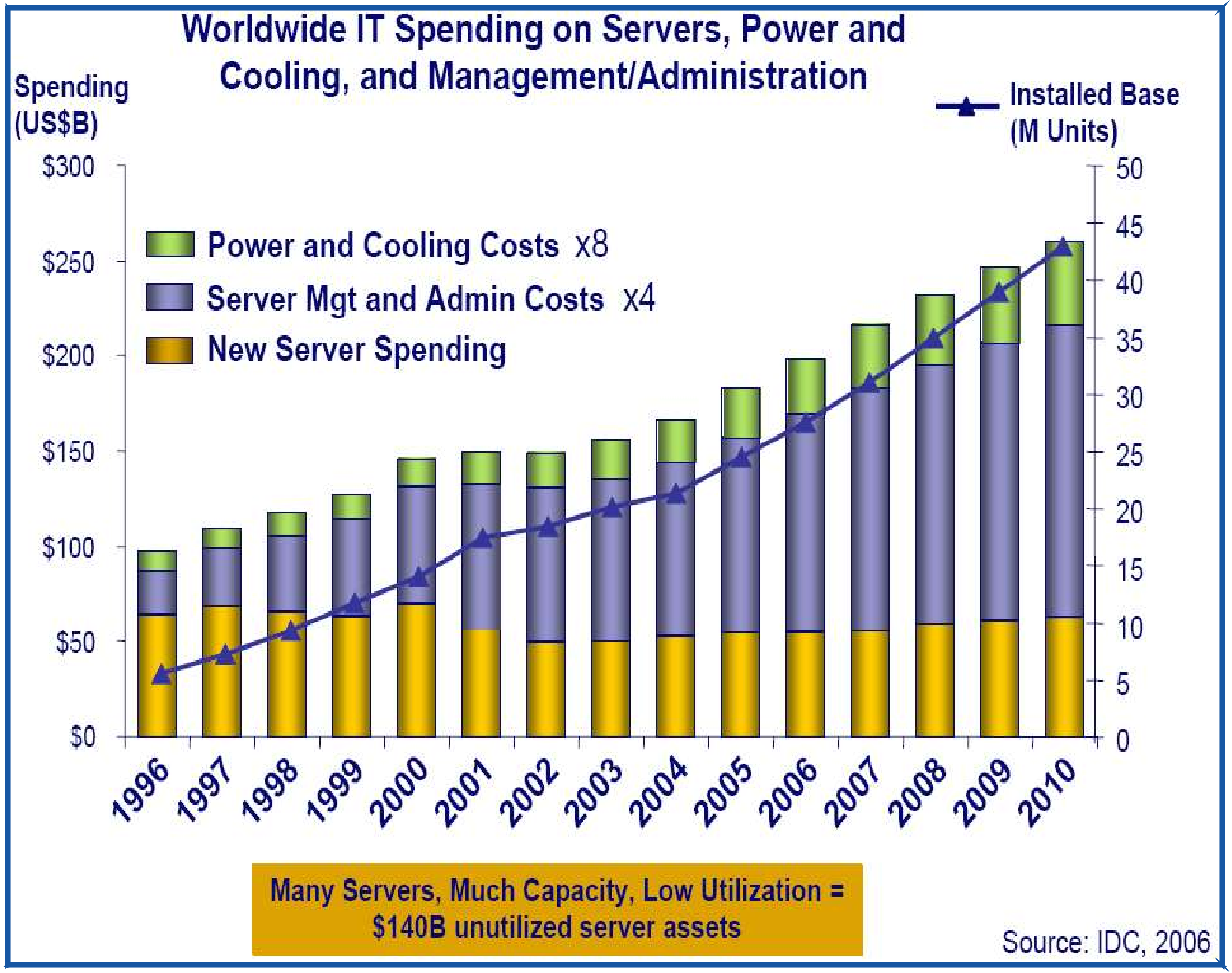
Graham White <gwhite@uk.ibm.com>
IBM Emerging Technology Services

Agenda (40 mins)

Virtualisation	~7 mins
Cloud Computing	~4 mins
Project Big Green	~4 mins
Security Enhanced Linux (SELinux)	~5 mins
Real Time	~10 mins
Questions & Answers	~10 mins

Virtualisation

IT Management Costs



Source: IDC, Virtualization 2.0: The Next Phase in Customer Adoption, Doc #204904, Dec 2006

Where Does the Energy Go?

The data centre energy challenge affects both the physical data centre and the IT infrastructure, typical usage shown below:

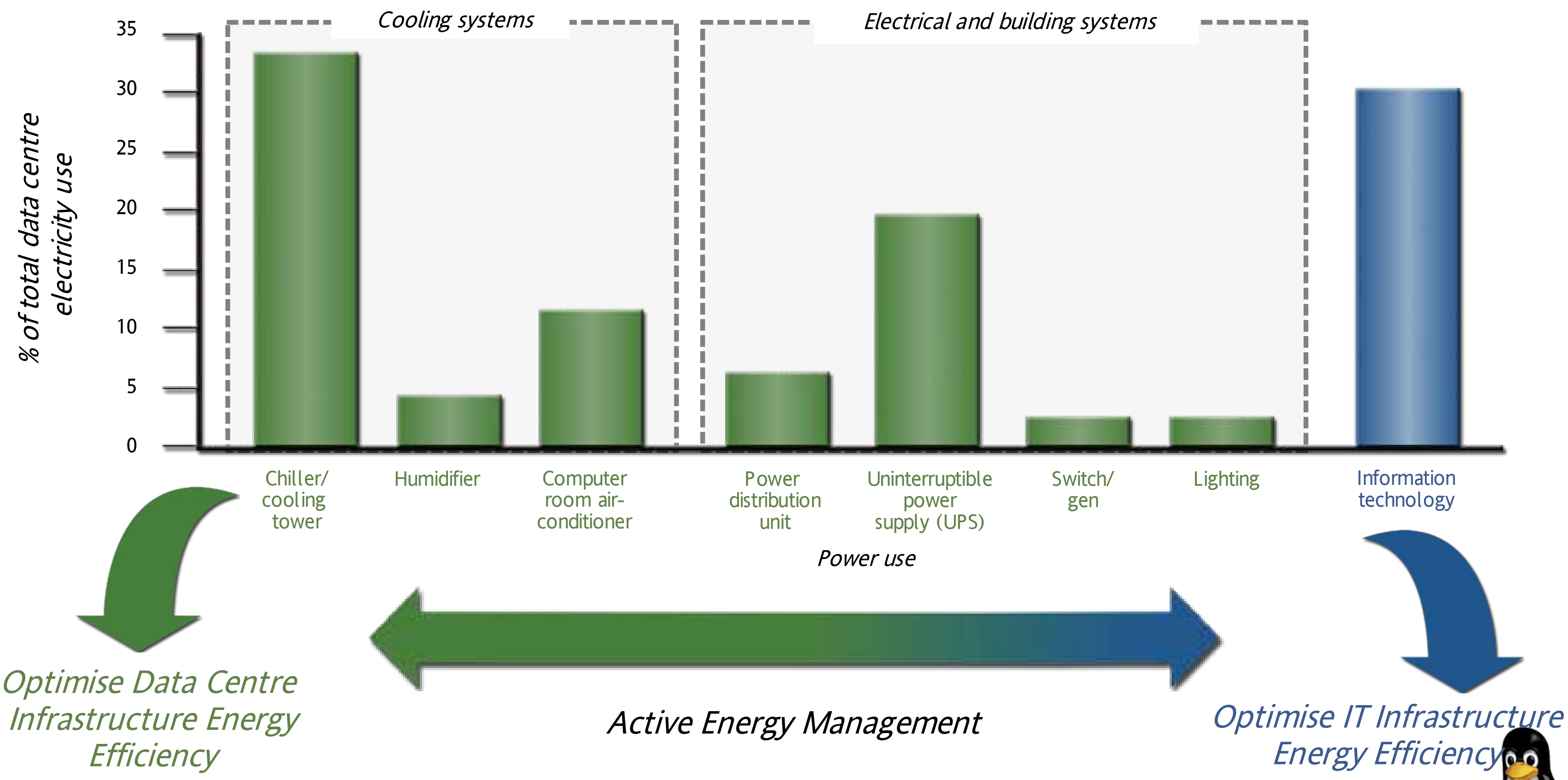


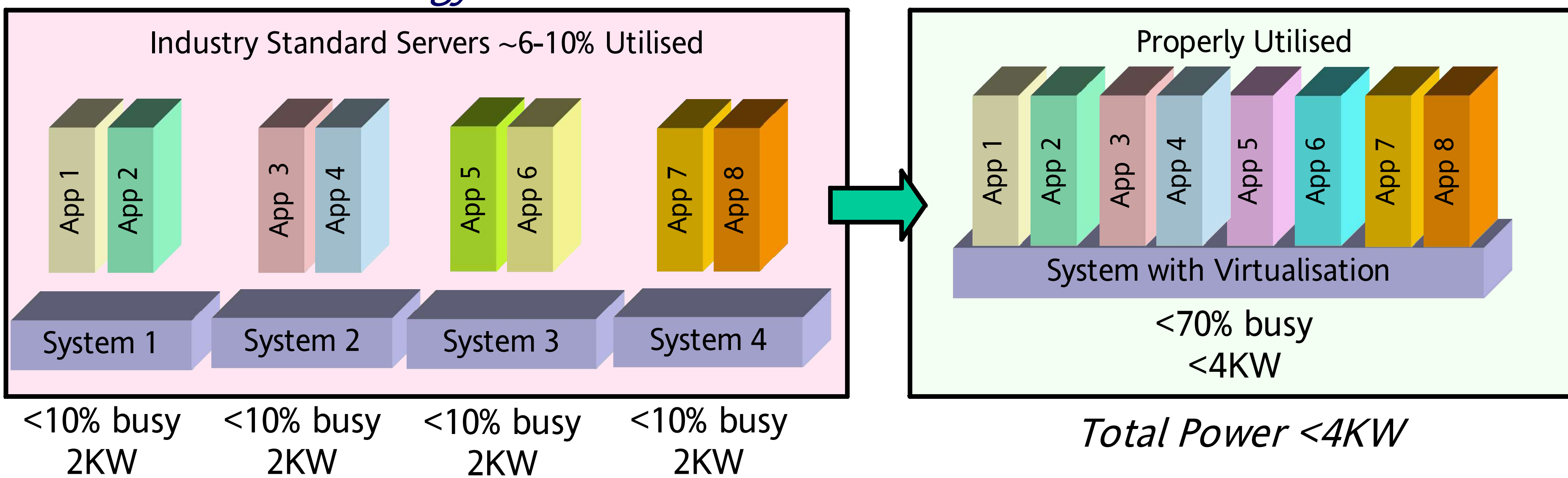
Chart and data source: American Power Conversion Corporation (APC) white paper, Implementing Energy Efficient Data Centres, #114, by Neil Rasmussen, 2006.

Virtualisation

Server Consolidation Can Reduce Energy Consumption

Energy Management Example –

Conserves Energy Without Loss of Performance



Total Power 8KW

- Server consolidation exploiting virtualisation is a very effective method for reducing energy costs
- IBM and our clients are seeing the benefits from virtualisation of servers and storage

Virtualisation

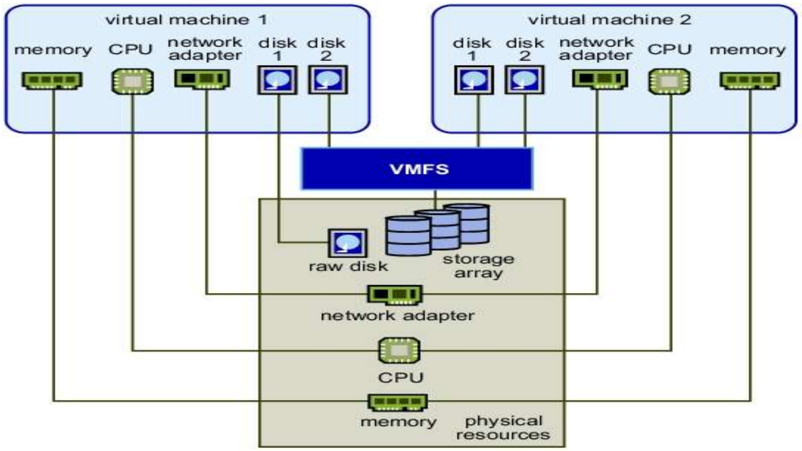
Benefits

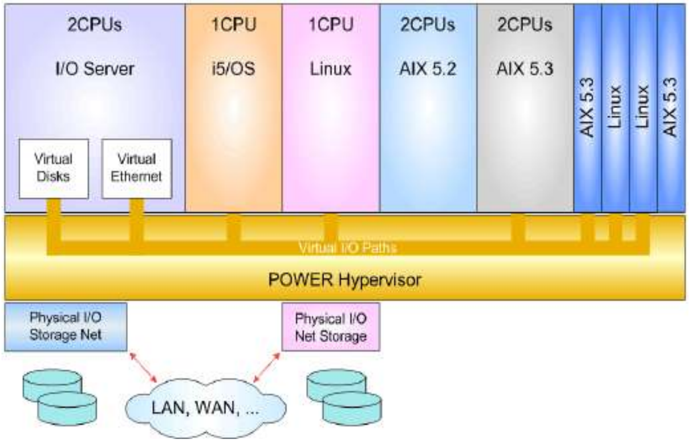
- IT Savings:
  - Improve equipment utilisation
  - Reduce hardware requirements

- Operational Savings:
  - Reduced space requirements
  - Power and cooling costs
  - Maintenance
  - Software support
  - Administrative costs

IBM Data Centre Case Study:

- 400+ Data Centres around the globe
  - 5% used for internal operations, 95% used for SO
  - 200K servers (~50/50 split between Unix and Wintel)
- Results:
  - Freed up physical equipment
  - Achieved 70% in operational savings
    - Space, power and cooling, maintenance, software support and personnel costs

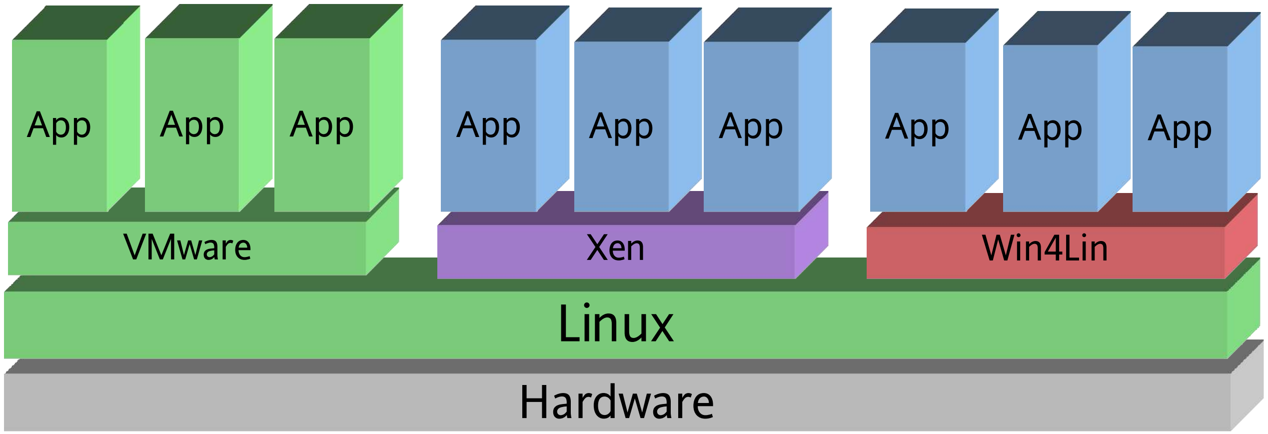




Linux

Virtualisation

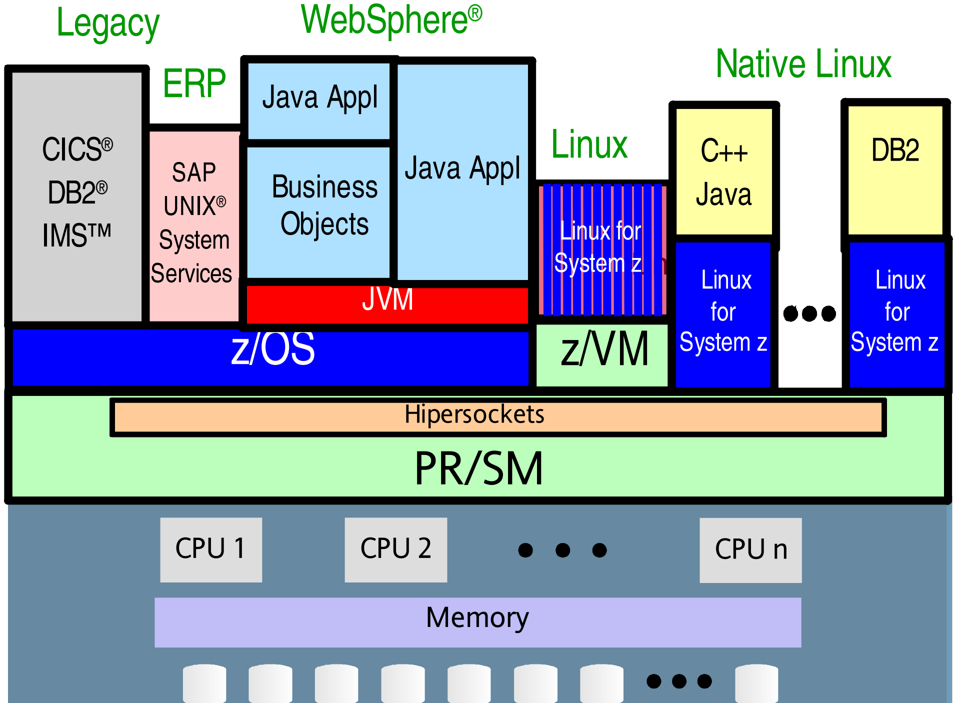
Some Linux Options



- Hardware assisted (e.g. Xen or KVM)
  - Uses AMD-V or Intel VT-x instructions in latest CPUs
- Full software virtual machine (e.g. VMWare or qemu)
  - VM software interrupts privileged calls
- Paravirtualisation (e.g. Xen)
  - Guest runs a kernel which knows it's being virtualised
- Shared-kernel based virtualisation (e.g. Linux-vserver or OpenVZ)
  - Low overhead, very quick to start/stop

Virtualisation

System z – Ultimate Virtualisation



- Utilisation often > 80%
- Handles peak workload utilisation of up to 100% without service degradation for high priority workloads
- Most sophisticated and complete hypervisor function available

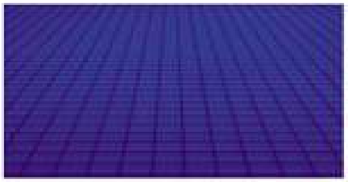
- Massive, robust consolidation platform; virtualisation is built in, not added on
- 100’s to 1000’s of virtual servers on z/VM
- Hipersockets for memory-speed communication, as well as Virtual Hipersockets via Guest LANs in z/VM
- ESCON channels and network support efficient sharing, I/O Passthru for fast I/O
- Most sophisticated and complete hypervisor function available
- Intelligent and autonomic management of diverse workloads and system resources based on business policies and workload performance objectives

Cloud Computing

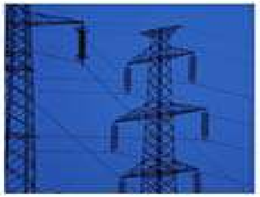
Cloud Computing
The History

The Evolution to Cloud Computing

Grid Computing
- Solving large problems with parallel computing
- Made mainstream by Globus Alliance



Utility Computing
- Offering computing resources as a metered service
- Introduced in late 1990s



Software as a Service

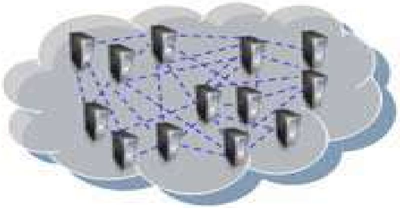
Network-based subscriptions to applications

Gained momentum in 2001

Cloud Computing

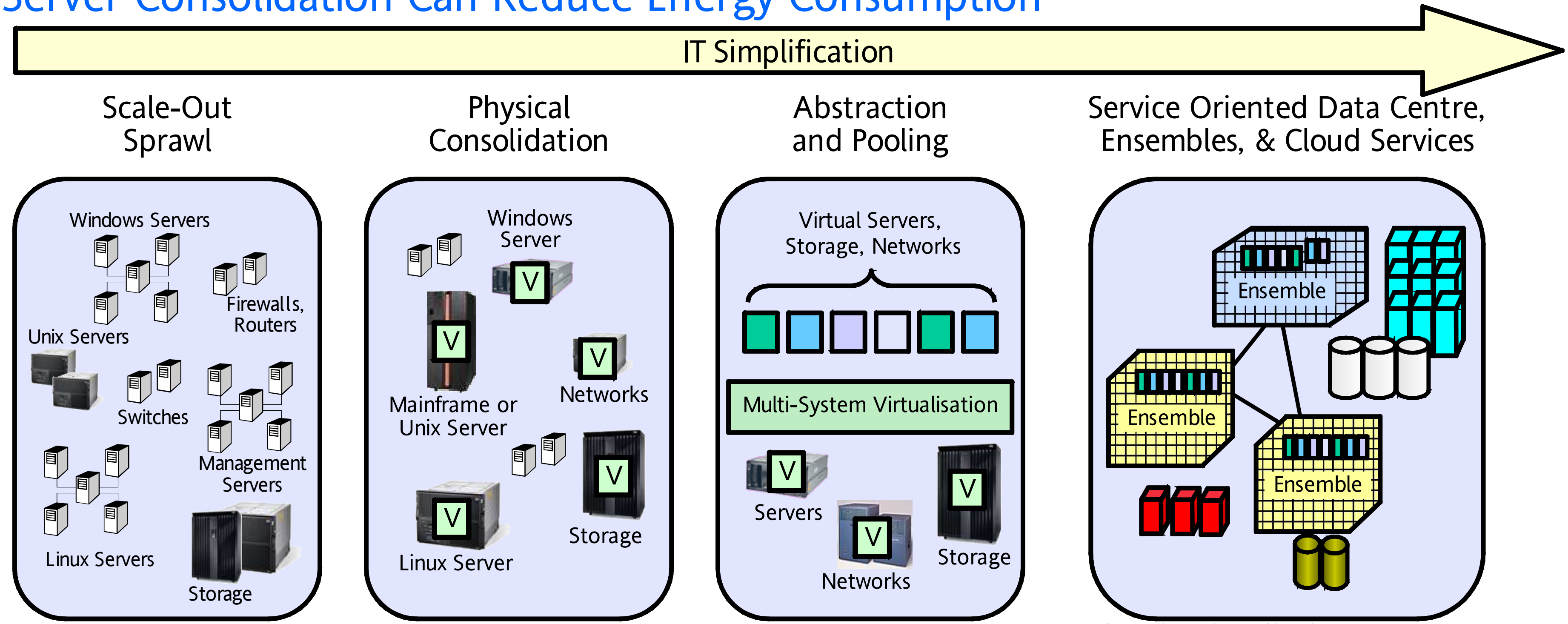
Next-Generation Internet computing

Next-Generation Data Centers



Cloud Computing

Server Consolidation Can Reduce Energy Consumption



Top IT Requirements:

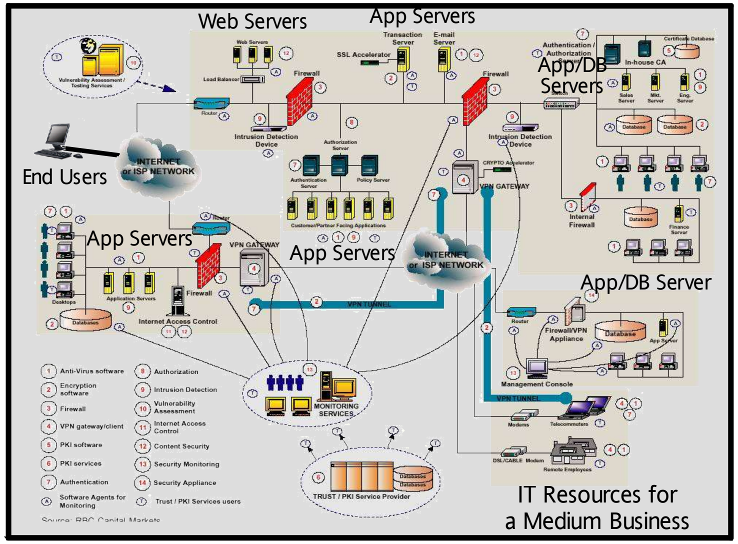
- Lower total costs
- Rapid application deployment, including self-service
- High availability, security, energy efficiency, utilisation
- Service oriented infrastructure

- Comprehensive virtualisation
- Integrated QoS management of availability, security, perf., …
- Hierarchical complexity hiding
- Self-service to resources & apps
- Request driven provisioning
- Automated app. on-boarding
- Appliances and SaaS support
- Cloud computing service model

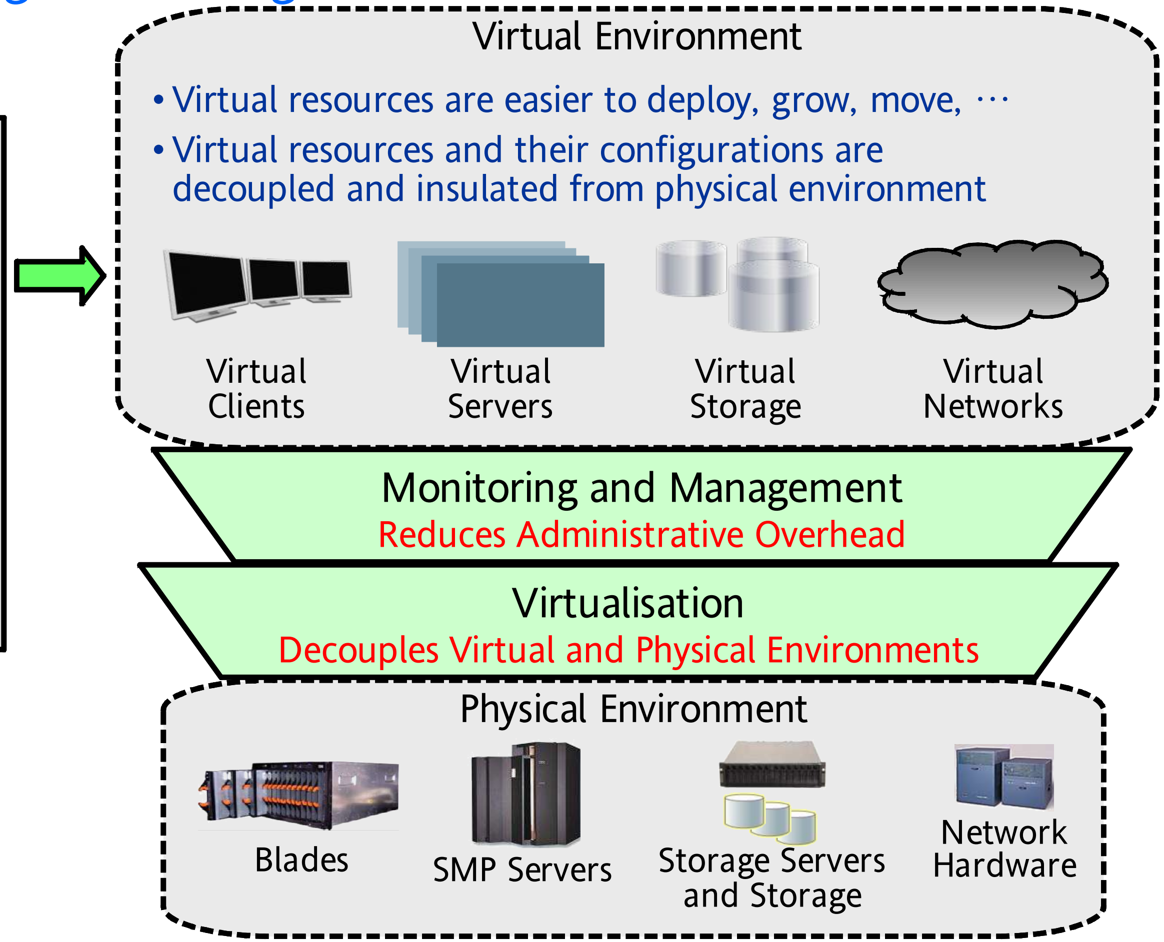
Cloud Computing

IT With Virtualisation, Monitoring and Management Software

IT Without Virtualisation



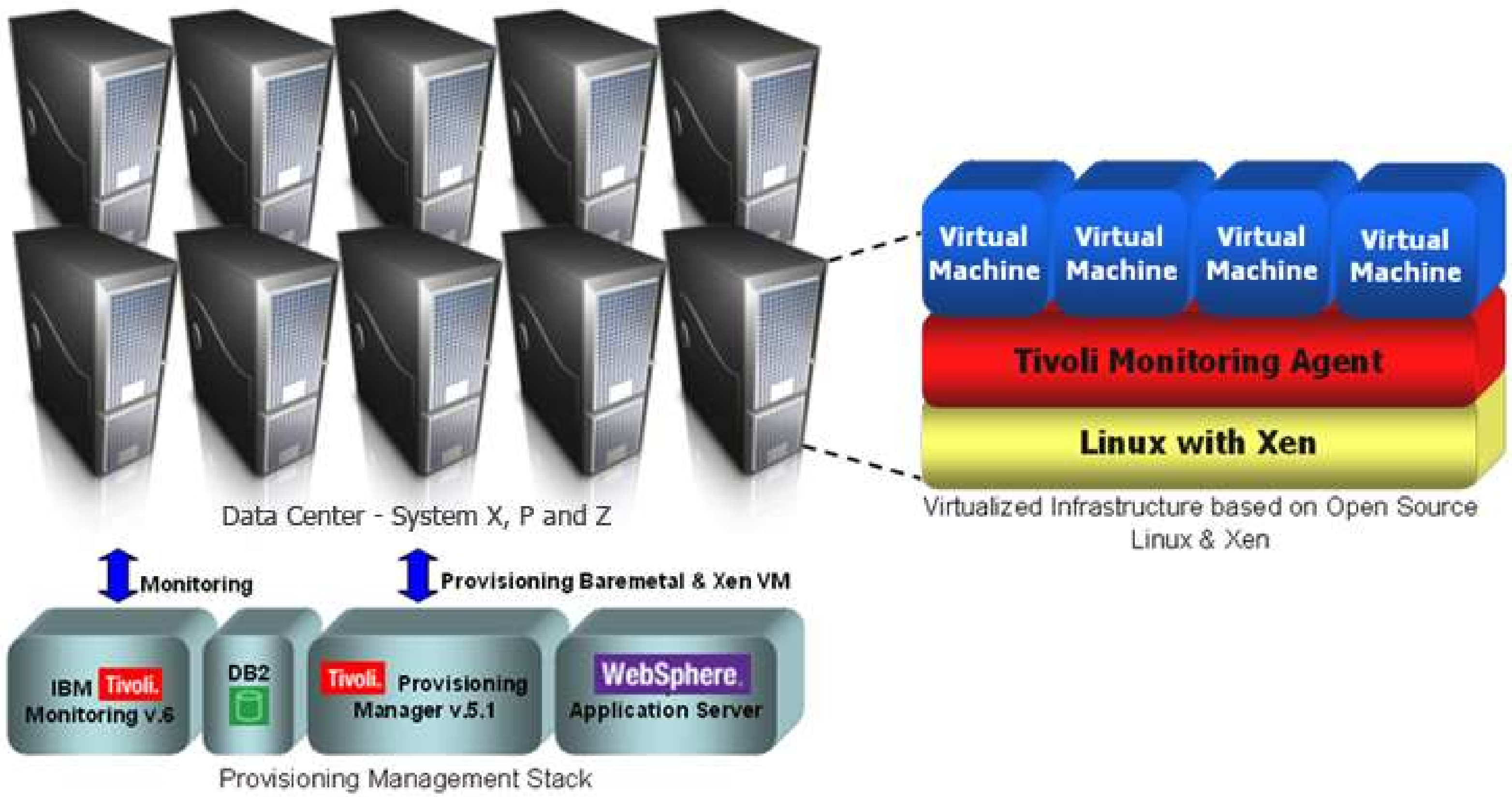
- Rigid configurations
- Fixed resources per server
- Low server utilisation
- Wasted energy and floor space
- New HW impacts SW assets
- Servers managed individually



✱Virtualisation technologies with new management software will significantly reduce IT costs and fulfill "on demand" / SOA needs.

Cloud Computing
The IBM Stack

IBM hardware and software combine with open source to make a Linux cloud



Project Big Green

Environmental Responsibility – Core Value at IBM

New Goal Announced!

Further extend IBM's early accomplishments by reducing CO_2 emissions associated with IBM's energy use 12% from 2005 to 2012 via energy conservation, use of renewable energy, and/or funding CO_2 emissions reductions with Renewable Energy Certificates or comparable instruments.

Awards & Recognition

BEST Workplaces for Commuters – FORTUNE 500 Top 20 2004, 2005, 2006

Energy Star – 1998, 1999, 2001

The Climate Group 2005

Climate Leaders (U.S. Environmental Protection Agency) – 2005

WWF climate savers – 2005

USEPA Climate Protection Award 1998 and 2006

Green Power Purchaser Award 2006

Environmental Efforts at Big Blue

Energy Star – Computer Program Charter Member 1992

WWF climate savers – Charter Member 2000

CCX Chicago Climate Exchange – Charter member 2003

SmartWay Transport Partnership (U.S. Environmental Protection Agency)

Pew Center on Global Climate Change – Business Environmental Leadership Council

Climate Leaders (U.S. Environmental Protection Agency) – Charter Member 2002

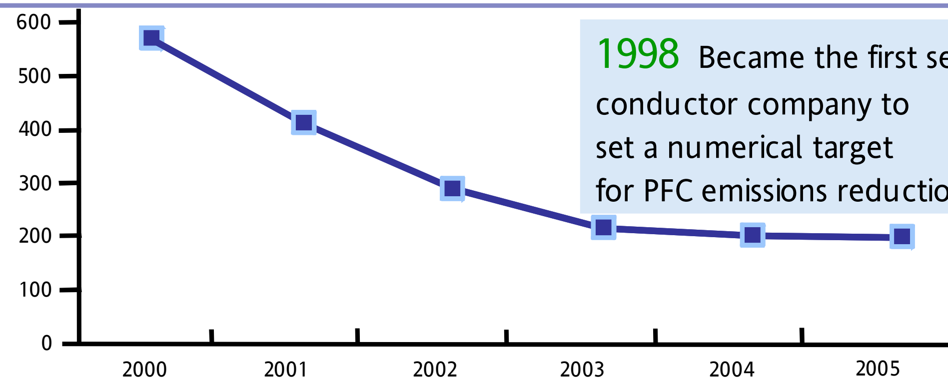
Climate VISION – 1605(b) voluntary emissions reporting since 1995

CARBON DISCLOSURE PROJECT – Since inception

Long History

40%

Between 1990 and 2005, IBM's global energy conservation actions reduced or avoided CO_2 emissions by an amount equal to 40% of its 1990 emissions.

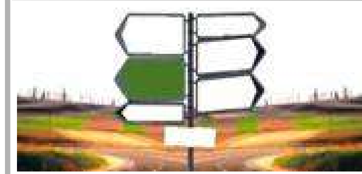


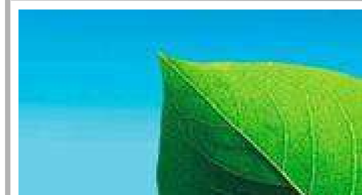
1998 Became the first semi-conductor company to set a numerical target for PFC emissions reduction

58%

Project "Big Green"
How are we doing it?

- IBM to reallocate $1 billion each year:
  - To accelerate "green" technologies and services
  - To offer a roadmap for clients to address the IT energy crisis while leveraging IBM hardware, software, services, research, and financing teams
  - To create a global "green" team of almost 1,000 energy efficiency specialists from across IBM
- Re-affirming a long standing commitment at IBM:
  - Energy conservation efforts from 1990 – 2005 have resulted in a **40% reduction in CO_2 emissions** and a **quarter billion dollars of energy savings**
  - Annually invest $100M in infrastructure to support remanufacturing and recycling best practices
  - *Will double compute capacity by 2010 without increasing power consumption or carbon footprint saving 5 billion kilowatt hours per year . . . equals energy consumed by Paris - "the City of Lights"*
- What "green" solutions can mean for clients:
  - For the typical 25,000 square foot data center that spends $2.6 million in power annually, energy costs can be cut in half
  - Equals the reduction of emissions from taking 1,300 automobiles off of the road



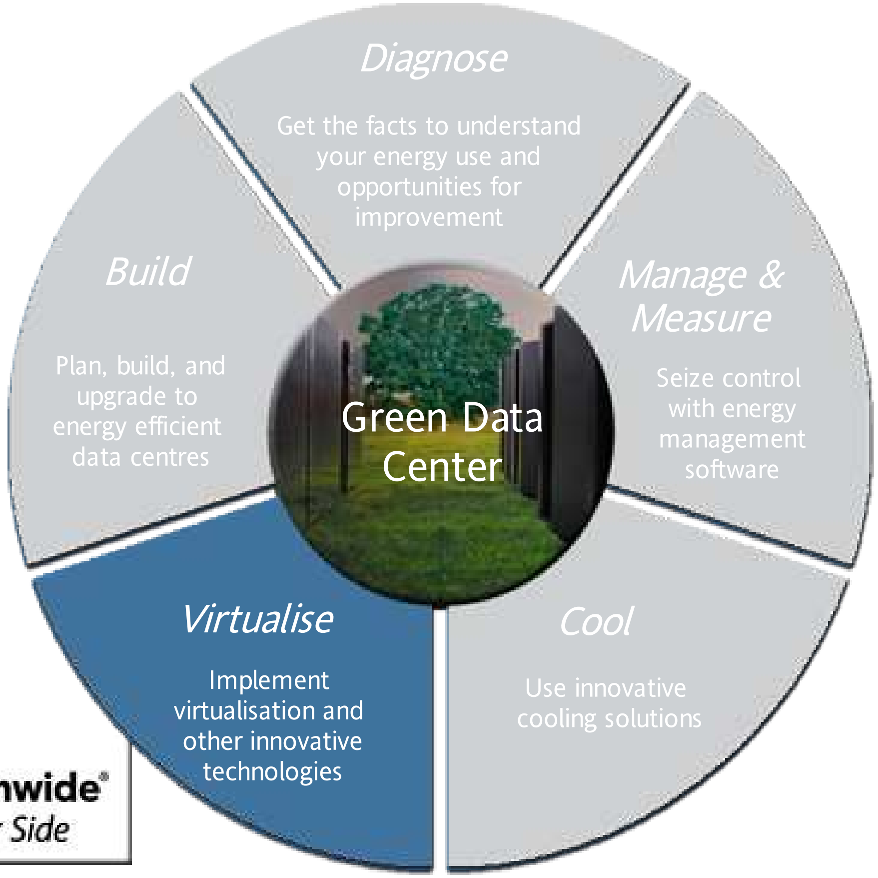


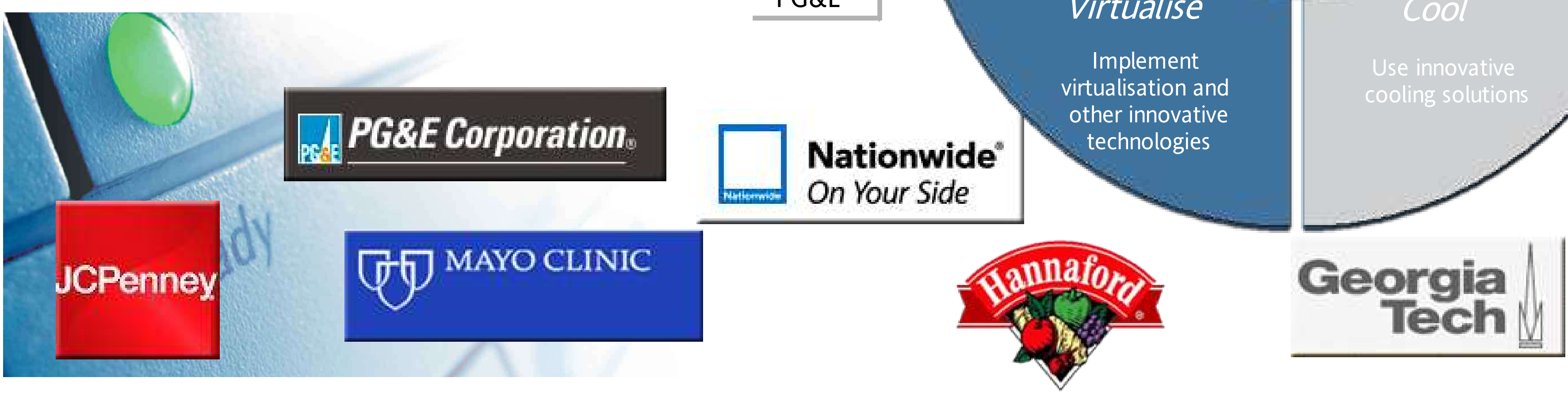
Project "Big Green"

What Others Are Saying

- IBM and our clients are seeing results from virtualisation of servers and storage

> *"Energy efficiency is the number one priority for PG&E as we work with our customers to meet our environmental goals. We're thrilled to partner with IBM to pilot energy efficiency innovations that can help our customers save money and protect the environment by further reducing their energy use."*
>
> Brad Whitcomb,
> VP, Customer Products & Services,
> PG&E





1. Gartner, *Toolkit: The Impact of Virtualization on Data Center Facilities*, G00148020, June 2007.

Project "Big Green"
Assessing the Need

Business

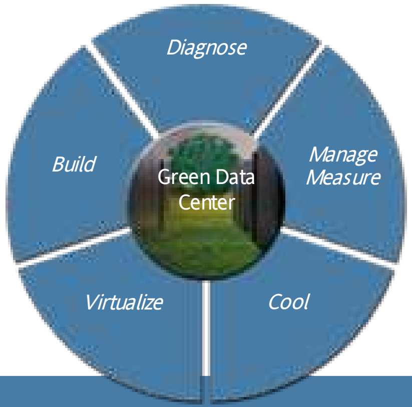
- Large number of distributed machines with low utilisation
- Compelling need that triggers action, examples:
  - Cost savings
  - Facilities issue (space, power, cooling)
  - Labour or environmental requirement
  - Regulatory requirement
  - Reliability, Availability or Security
  - Provisioning requirements
  - Business resiliency, Throughput or Performance need
  - Maintenance or upgrade requirement

Technical

- Applications with:
  - peaks at various times
  - a low peak as a group
  - considerable I/O, throughput
  - data sharing and queries
  - heavy synchronisation
- Workloads that migrate well
  - Linux, UNIX, HPUX, Solaris, AIX, Certain Windows Operating Systems
  - Domino, e-mail solutions, LDAP
  - WebSphere Process, Portal, MQ, WAS
  - SAP, .NET Applications
  - DB2, Lotus, Rational, Tivoli, Oracle, Web Logic, New CICS, IMS Connect for Java
  - Network
  - Data Warehouse

Project “Big Green”

Benefits from Going Green



	From	To
Financial	Rising global energy prices Squeeze on IT budgets Constraints on IT growth	Ability to accurately view baseline energy cost Cost savings from more efficient energy use Reduce power and cooling issues as inhibitor to business growth
Operational	High density server systems Exploding power & cooling cost Aging data centers	More computing performance per kilowatt Shift to have more energy used by IT than physical infrastructure Extend the life of existing data centers
Environmental	Corporate social responsibility Limited “green” public image Improve employee moral	Meaningful energy conservation and reduced CO_2 emissions Improve “green” public image Positive impact linking corporate social responsibility and personal values

Project “Big Green”

Linux Energy Management

- Idle CPU power management
- Timer migration from idle CPUs
- Scheduler heuristics for power aware SMP load balancing
- Power optimised system userspace and subsystems (via powertop analysis of frequent wakeups)
- Stable Dynamic Voltage and Frequency scaling (DVFS) in kernel (cpufreq)
- Support for low power standby CPU states where applicable

Active Energy Manager

- Monitor, measure and control their energy usage
- Returns true control of energy costs to the customer
- Single view of the actual power usage across multiple platforms in their infrastructure
- Supports across a large spectrum of IBM Systems
- Part of IBM Director and available on all systems, P, I, X, Storage and Z.

Security Enhanced Linux (SELinux)

Security Enhanced Linux

IT Security is Still Important

✱Recent News:

– *Foreign hackers have compromised cash register terminals at 11 Dave & Buster's restaurants around the United States. The scheme resulted in losses of some $600,000*

– *Source: SC Magazine, May 2008*

– *According to various news reports, a hacker has exposed the personal information of about six million Chilean people. The hacker, known as Anonymous Coward, is reported to have penetrated government and military servers to steal data, including ID card numbers, addresses, telephone numbers, emails and academic records.*

– *Source: SC Magazine, May 2008*

✱Cost from Security Breaches:

– *The average security breach can cost a company between $90 and $305 per lost record, according to a new study from Forrester Research. The research firm surveyed 28 companies that had some type of data breach.*

– *Source: InformationWeek, April 2007*

Security Enhanced Linux
What is it?

- SELinux is
  - Security Enhanced Linux
  - Started by the National Security Agency (NSA) as part of its Information Assurance mission
  - An implementation of Mandatory Access Control (MAC) for Linux
  - Provided as part of the 2.6 Linux kernel through the use of Linux Security Modules (LSM)
  - Shipped with Red Hat and some other non-Enterprise distributions
  - Free, re-distributable and open source
  - Designed to provide absolute security and be a complete solution for MAC

- SELinux *is not*
  - A Linux distribution

Security Enhanced Linux

Contrasting Traditional Security and SE Linux

Traditional Security

- Discretionary Access Control (DAC)
  - The operating system provides accounts for user login (including super user such as root)
  - Users are permitted access to resources based on ownership
  - Users may grant permission (directly or indirectly) to other users
  - Some resources may be labelled as world write/readable
- Additional security mechanisms available on some systems
  - Access Control Lists (ACLs)
  - Tokens e.g. AFS, GFS
  - File system security (ext2 immutable bit)
- Based on a black-list approach
  - Allow all access, except for that which has been specifically restricted

SE Linux

- Mandatory Access Control (MAC)
  - Security decisions allowed by DAC are further examined by a MAC policy
  - System wide security policy
  - Fine grain control over all system resources
    - Files, processes, network sockets and interfaces, etc
  - Security policy is separated from enforcement
    - Extremely secure
    - System can be administered locally but with corporate control over policy
  - Principle of least privilege can be applied system-wide
- Based on a white-list approach
  - Deny all access, except for that which has been specifically allowed

Security Enhanced Linux
Novell's AppArmour

- SELinux
  - The only complete solution
  - Only available with Linux
  - Community and distribution support
- AppArmour
  - Designed, written, maintained and supported by Novell
  - Only available with the SuSE Linux
  - Provides a sub-set of security functionality in SELinux
- Other Research Operating Systems and non-commercial systems

- SELinux vs AppArmour (in brief)
  - "SELinux is designed to be a complete Mandatory Access Control solution"
  - "AppArmour is designed to be simple to understand, configure, and operate"
  - The result is SELinux is more difficult to configure and understand but the extra effort provides a more compete and secure solution
  - Both use the same kernel security modules in Linux, known as LSM (Linux Security Modules), to interface to the kernel security decision making system

Security Enhanced Linux

How MAC Security Helps

- In a 2007 E-Crime Watch survey
  - Approximately 32% experienced e-crime between July 2006 and June 2007
  - Approximately 13% experienced theft of proprietary data (i.e. customer records, financial records, etc.)
  - The implications include lost taking systems down, forensics, emergency fixes, monitoring, testing, system re-start, analysis, etc.
- With MAC
  - Less time is required for forensics (security logs virtually incorruptible)
  - Access to system resources is confined and contained such that the resource restrictions imposed by the policies are virtually unbreakable by the vulnerable application or by the attacker

"We consider Mandatory Access Control to be a key enabling technology to aid government and businesses alike, in being confident they can deliver more services, more quickly and with better function, without compromising security."

-- Dr. Steve Marsh, Intelligence and Security Advisor, UK Cabinet Office

Real Time

Real Time Computing

What is it?

- Real-time computing – Computation that fails if it does not complete within a specified deadline.
  - *"Real-time computing is the study of hardware and software systems which are subject to constraints in time. In particular, they are systems that are subject to deadlines from event to system response; that is, the computation can be said to fail if it does not complete in the time-period after the event before the deadline relative to the event. These deadlines are independent of system load."*
    - *Source: www.en.wikipedia.org "Real-time computing"*
  - *"Describes an application which requires a program to respond to stimuli within some small upper limit of response time (typically milli- or microseconds). Process control at a chemical plant is the classic example. Such applications often require special operating systems (because everything else must take a back seat to response time) and speed-tuned hardware. "*
    - *Source: Foldoc.org "Real-time"*

Real Time
Hard vs. Soft

- Real-time computing – Computation that fails if it does not complete within a specified deadline
  - A "hard" real-time application has tasks which have hard, firm deadlines for completion of their computation, typically on the order of a few microseconds. Missed deadlines cause system failure (at best) or life-threatening catastrophe (at worst).
  - Examples:
    - Federal mission critical and Financial trading
    - Industrial automation (robot picking up an object from a moving belt)
    - Aircraft flight control
    - Medical devices/imaging
  - A "soft" real-time application can tolerate some missed deadlines and does not typically require hard scheduling guarantees from the OS.
  - Examples:
    - Point of sale system
    - Consumer devices (PDA, phones, etc…)
    - Massive multiplayer game servers
    - Media display/playback

Real Time

Why is it important?

- Real-Time Solutions are Predictable and Provide Guaranteed Response Times

- Predictable Application Response Time Has Broad Industry Appeal
  - Defense
  - Energy & Utilities
  - Financial Services
  - Gaming
  - Industrial Automation
  - Telecommunications

Real Time
Problems and Difficulties with Commercial Usage

- Specialised Hardware
  - Black box, not flexible
- Specialised Real Time Operating Systems (RTOS)
  - Closed source and/or proprietary
  - Linux™ not widely used for real-time systems due to
    - *Unpredictable scheduling*
    - *Low timer resolution (10 ms granularity)*
    - *Non-preemptible kernel*
- Applications written in C, C++, Ada
  - Required specialised skills, and applications were not reusable or portable
  - Java™ not widely used for real-time systems due to
    - *Regular Java Threads*
    - *Garbage Collection*
    - *Class Loading*
    - *Just-in-time (JIT) Compiling*

Real Time
What has changed?

Real Time Stack that is Open and Flexible

- x86 Architecture and Linux
  - *Control over architecture*
  - *Avoid vendor lock-in*
  - *Facilitate interoperability across heterogeneous systems*
  - *Real Time Linux expedites offerings in marketplace*

- Java Programming Language
  - *Access to broader skill set than C, C++, ADA*
  - *Increased programmer productivity*
  - *Expedited Real Time application development and support*
  - *Expedited innovation through reuse and collaboration*
  - *Massive community of ISVs*

Real Time

The Power of Linux and Java Combined

WebSphere® Real-Time (WRT)

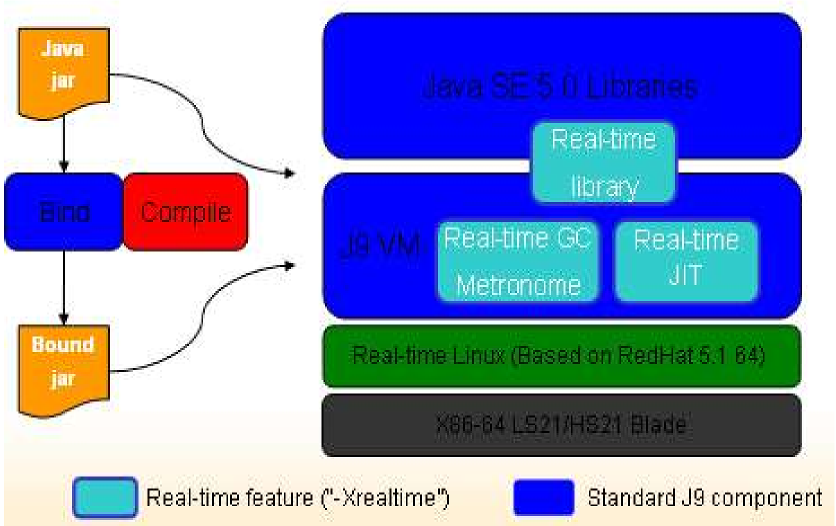
- Java 2 Standard Edition *
- Metronome Garbage Collector (GC) *
- Compilation Strategies for Real-Time (AOT, JIT) *
- Real-Time Specification for Java (RTSJ – JSR 1)

Standard or Real Time Linux

- High resolution time and timers
- Fully pre-emptible kernel
- Threaded interrupt handlers
- Priority inheritance and fast user-space mutexes
- Symmetric Multiprocessing (SMP) RT scheduling

Select IBM System X Hardware

- Enhancements to optimise real-time workloads



* Feature available for both soft real-time and hard real-time latencies.

Real Time

Available on Selected IBM System X Hardware

- LS21 and HS21 XM Blades have been optimised to support real-time workloads
  - System Management Interrupts (SMIs) are used to perform a variety of tasks at the CPU level
    - Report fatal and non-fatal hardware errors
    - Perform power management (thermal throttling, power capping)
  - SMIs introduce latency which are hard to detect
  - SMI remediation:
    - Moves non-fatal event handling to the OS
    - BIOS handles only fatal events
    - BMC no longer requests to throttle the CPU
    - New OS service “ibm-prtm” manages entering and exiting SMI-free state

Real Time

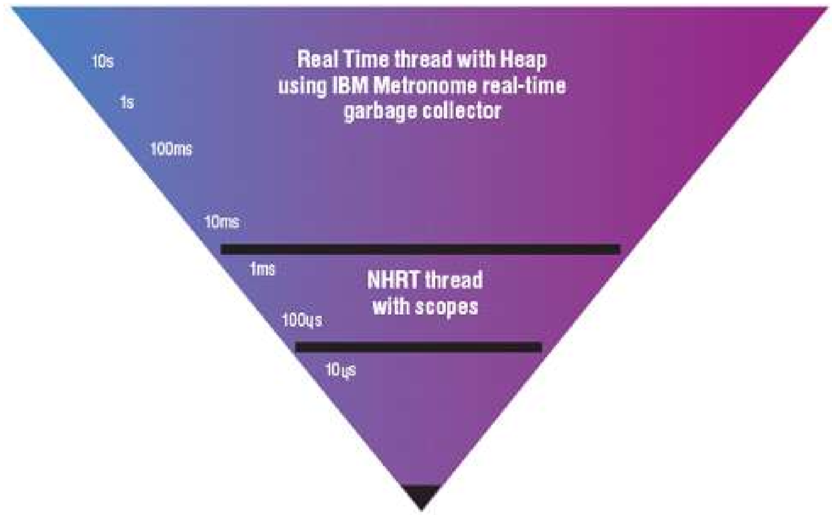
Real Time Linux Extensions

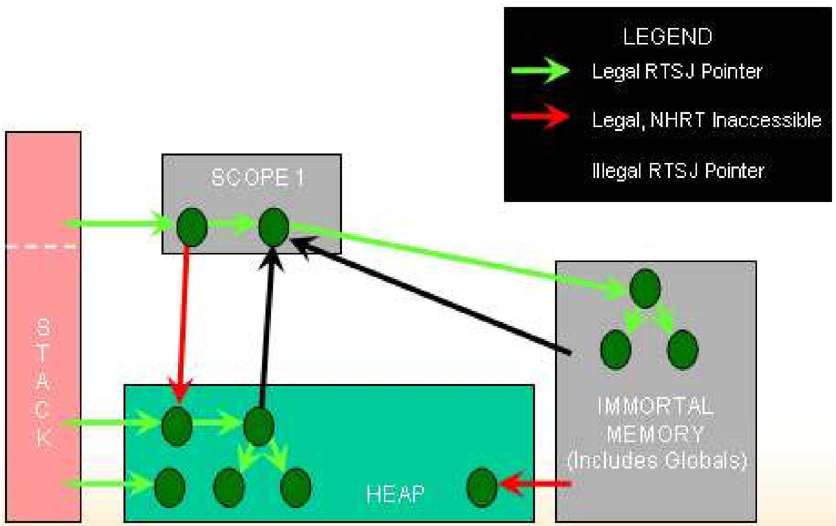
- High resolution time and timers
  - High-resolution timers provide single nanosecond granularity
  - Timers can now expire within a few microseconds of each other
- Fully pre-emptible kernel
  - Kernel spin-locks replaced with mutexes for higher levels of pre-emption
- Threaded interrupt handlers
  - Allows real-time processes to be prioritised against interrupt handler processes
- Priority inheritance and fast user-space mutexes
  - Kernel avoids priority inversion by raising priority of locking process to that of the highest priority process waiting for the lock
  - Fast-user space mutexes (futexes) help reduce regular mutex overhead
- SMP real-time scheduling
  - Allows real-time processes to be scheduled against all processor run queues

Real Time

IBM Java Implements the Real Time Java Specification (RTSJ)

- Thread scheduling
  - "RealtimeThread" allows specification of scheduling parameters
  - Fixed priority scheduling and additional priority settings





- Memory Management
  - Partitioned, non-garbage collected memory spaces
  - Allows special threads to run at higher priority than GC
  - Lower latency can be achieved using standard RTSJ scoped memory techniques for No Heap Real-time Threads (NHRT)

Real Time

IBM Java Metronome Garbage Collector

- Unique technology from IBM T.J. Watson Research
  - Garbage collection is scheduled as just another periodic real-time task
  - Provides bounded pause times as small as 1ms and a minimum utilisation level for application tasks
- Enables the use of off-the-shelf Java code
  - No need for specialised allocation schemes outside the Java heap
  - Greatly simplifies real-time application development
  - Enables complex real-time applications through easier composition

Figure 1. Traditional GC pauses

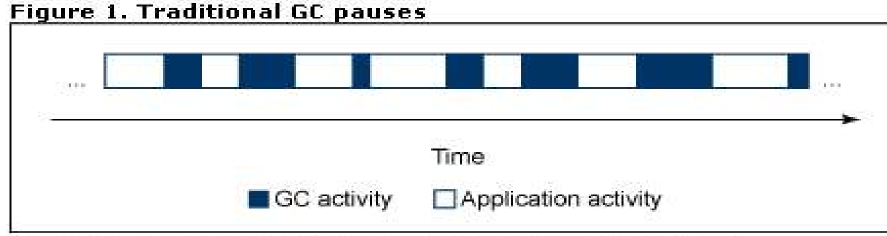


Figure 2. Short pause times but little application time

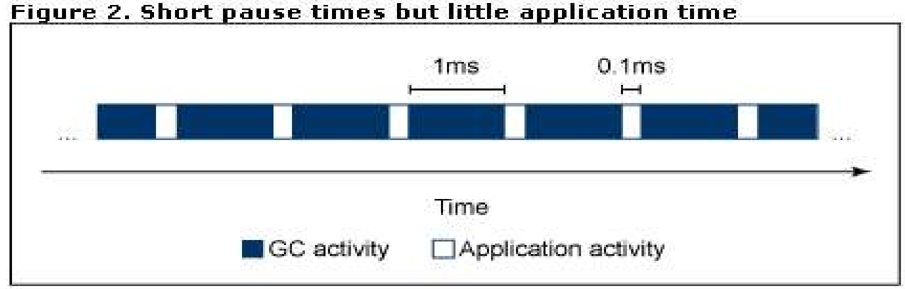
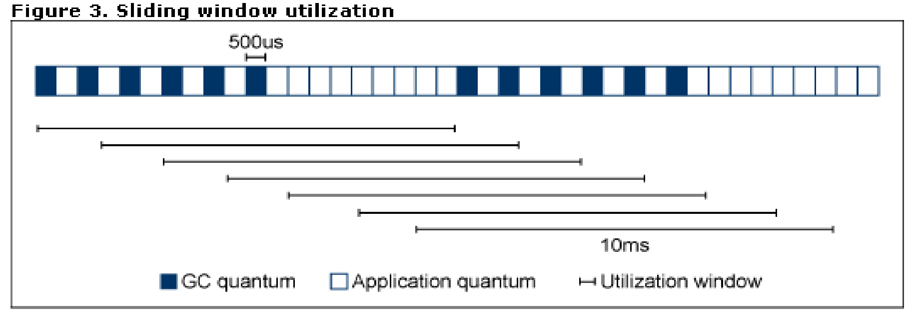


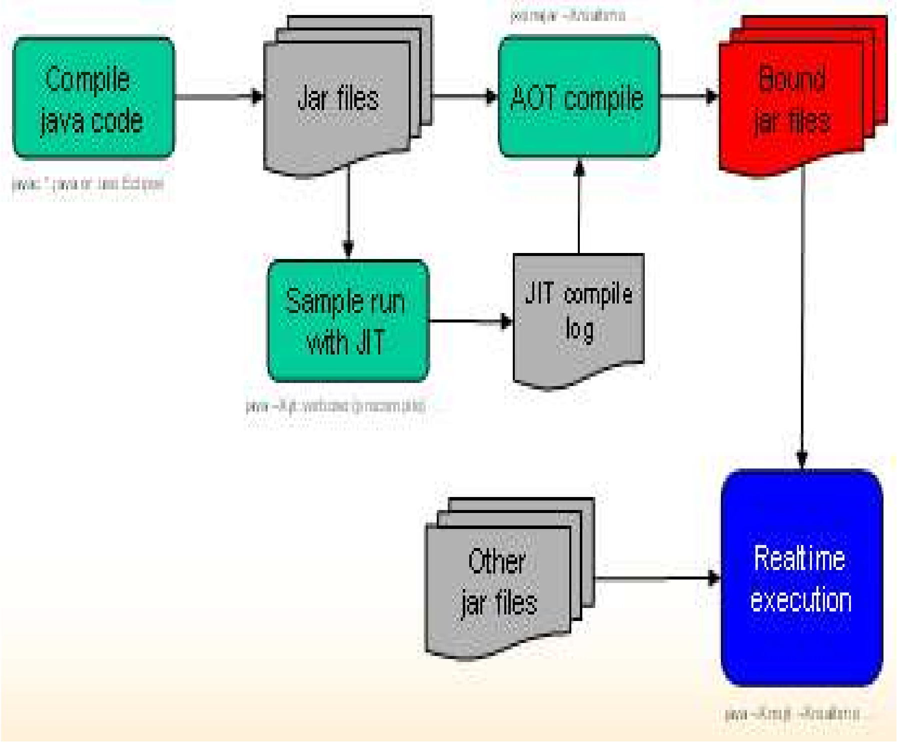
Figure 3. Sliding window utilization



Real Time

IBM Java Compilation Strategies for Real Time

- Compilation in J9 is dynamic by default
  - Code may be interpreted or compiled selectively through JIT, but JIT can run at unpredictable times and have unpredictable (although almost always positive) effect on performance
- For real-time, there are multiple choices
  - Ahead-of-time (AOT) compilation (much better than interpreted performance)
  - User-controlled JIT (better than AOT performance, controlled via API)
  - JIT-at-low-priority (best achievable performance, runs on low priority thread)



Real Time

Summary of Real Time Computing with IBM

- J9 (WRT) *
  - IBM-authored virtual machine used in all IBM products and on all platforms
  - Leadership performance, scalability and reliability
- Metronome (WRT) *
  - Real-time garbage collection with 1ms worst case pause time and providing assured minimum application CPU utilisation
- Optimising compilation (WRT) *
  - Static (or ahead-of-time) compilation for predictable performance
  - Dynamic (or just-in-time) compilation for best performance (predictable under controlled conditions)
- RTSJ (WRT)
  - Fully compliant with the latest version of JSR 1 (1.0.1b)
  - Includes fixed priority scheduling, priority inheritance, asynchronous event handling, scoped and immortal memory management
- Linux
  - RedHat or Novell *
  - Updated (open source) kernel for real-time operation
- Select IBM System X Hardware
  - With modifications to eliminate firmware induced delays

* Feature available for both soft real-time and hard real-time latencies.

Real Time

Real Time Evaluation Programme

- IBM has established a real-time evaluation program that allows customers to experiment with IBM real-time technology on Linux at no charge. The evaluation program consists of WebSphere Real Time (WRT) for soft and hard real-time workloads and access to Developerworks source code with documentation.

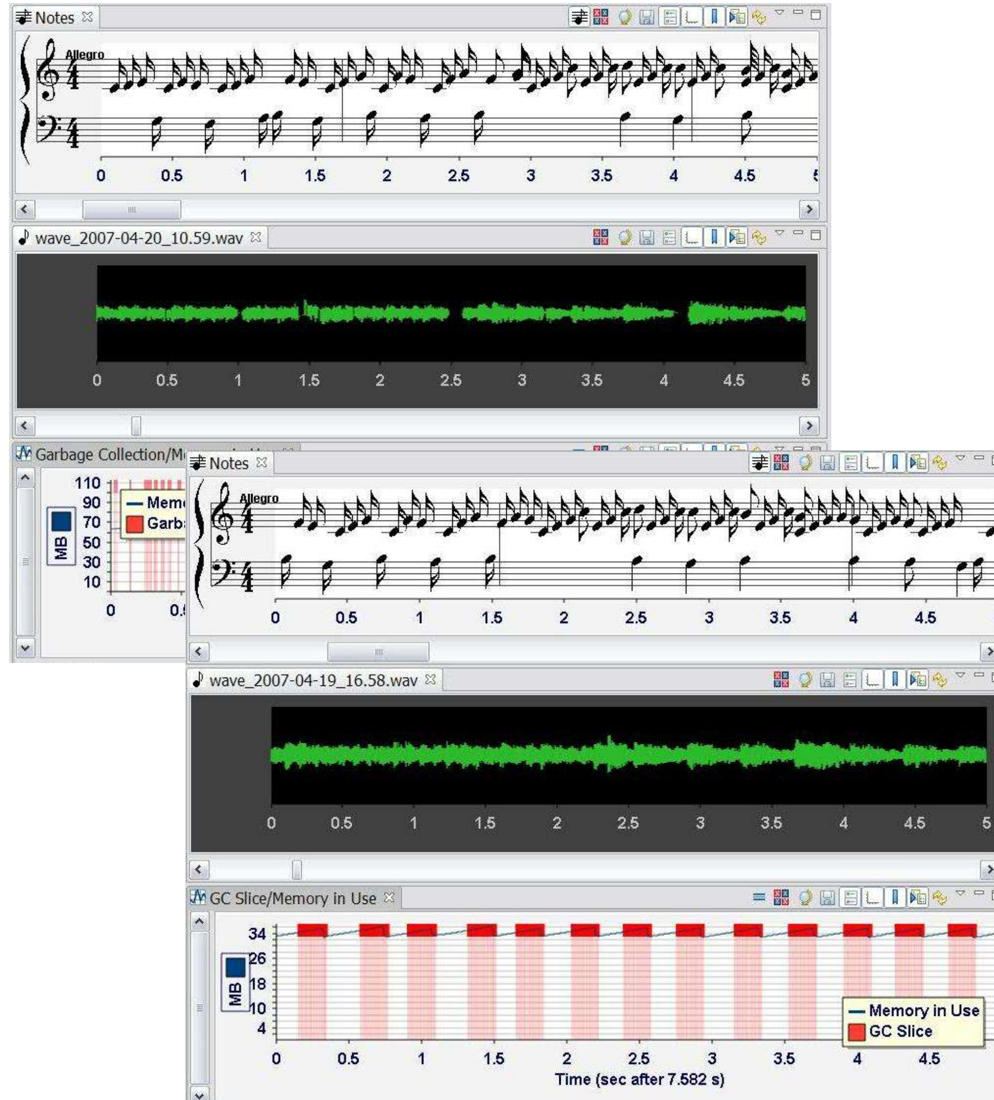
- For hard real-time workloads, IBM can loan one system with Real Time Linux pre-installed, for a period of 30-60 days upon signing a loaner agreement.

- Support is available via phone and/or email during normal working hours during the evaluation period. Response times may vary but our goal is to respond to all requests within 1 business day.

Real Time

Harmonicon Demo (Standard Java)

- The scenario
  - You'll listen to music play on a Java-based player.
  - The player will first run on a standard Java Virtual Machine without any real-time features enabled. As the music plays, garbage collection will take place sporadically.
- What will I see?
  - You will notice that sometimes when garbage collection occurs, the music gets distorted. Such distortions can be observed by gaps in the music (the green graph) which align with garbage collection interrupts (the red lines).



Real Time

Harmonicon Demo (Metronome Garbage Collector)

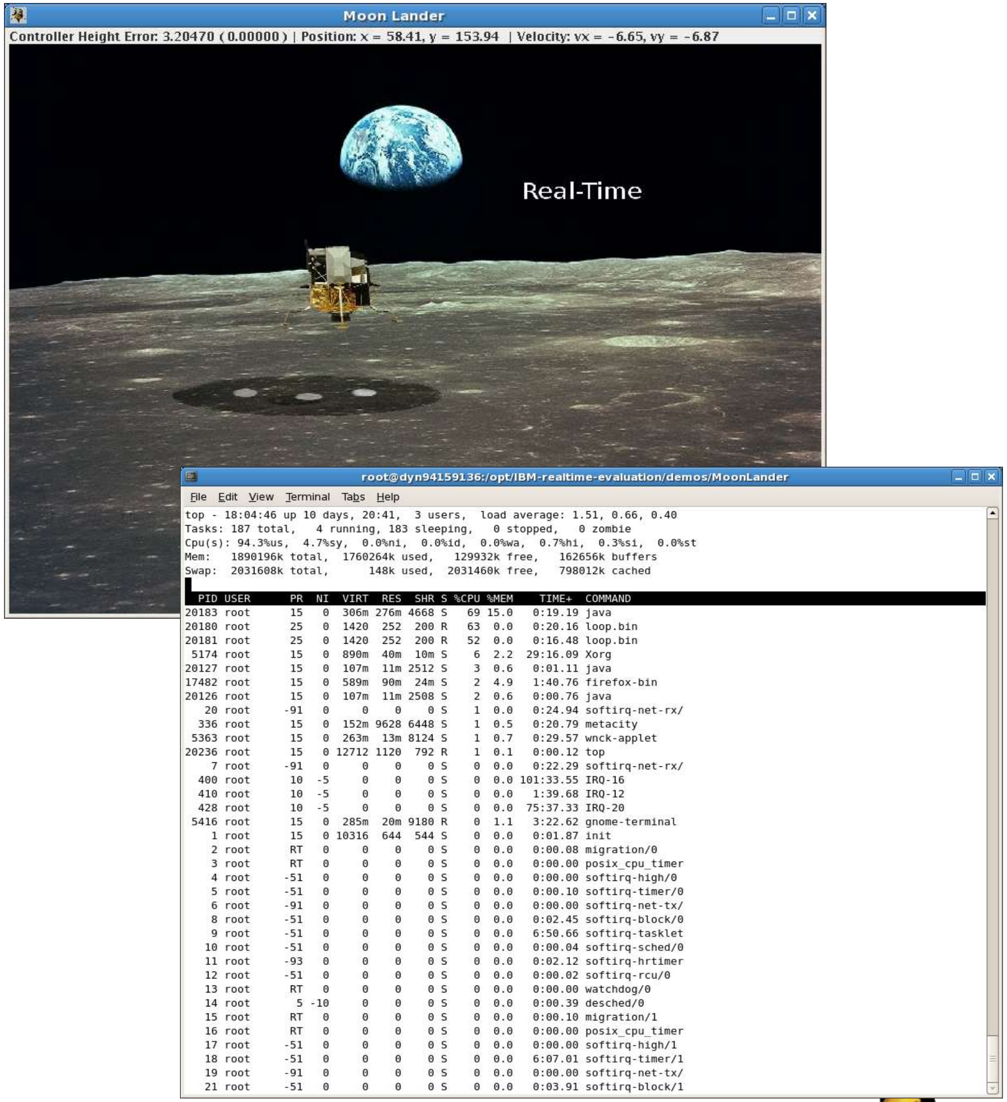
- The scenario
  - You'll listen to music play on a Java-based player.
  - The same Java-based player will be used to play the same music, except that this time Metronome will have been configured so garbage collection will occur more predictably and for shorter periods of time.
  - A real-time system is used this time.
- What will I see?
  - You will notice that the music plays without distortion. Garbage collection will still take place but cleanup will be done predictably and without interfering with the music.

Real Time

Moon Lander Demo (Standard Java)

- The scenario
  - You'll see the video of a lunar module landing.
  - In the simulation, the coordinates for the lunar module's landing area are transmitted from Earth, and the round trip time is used to estimate actual from expected trajectory.
  - A non real-time system is used.

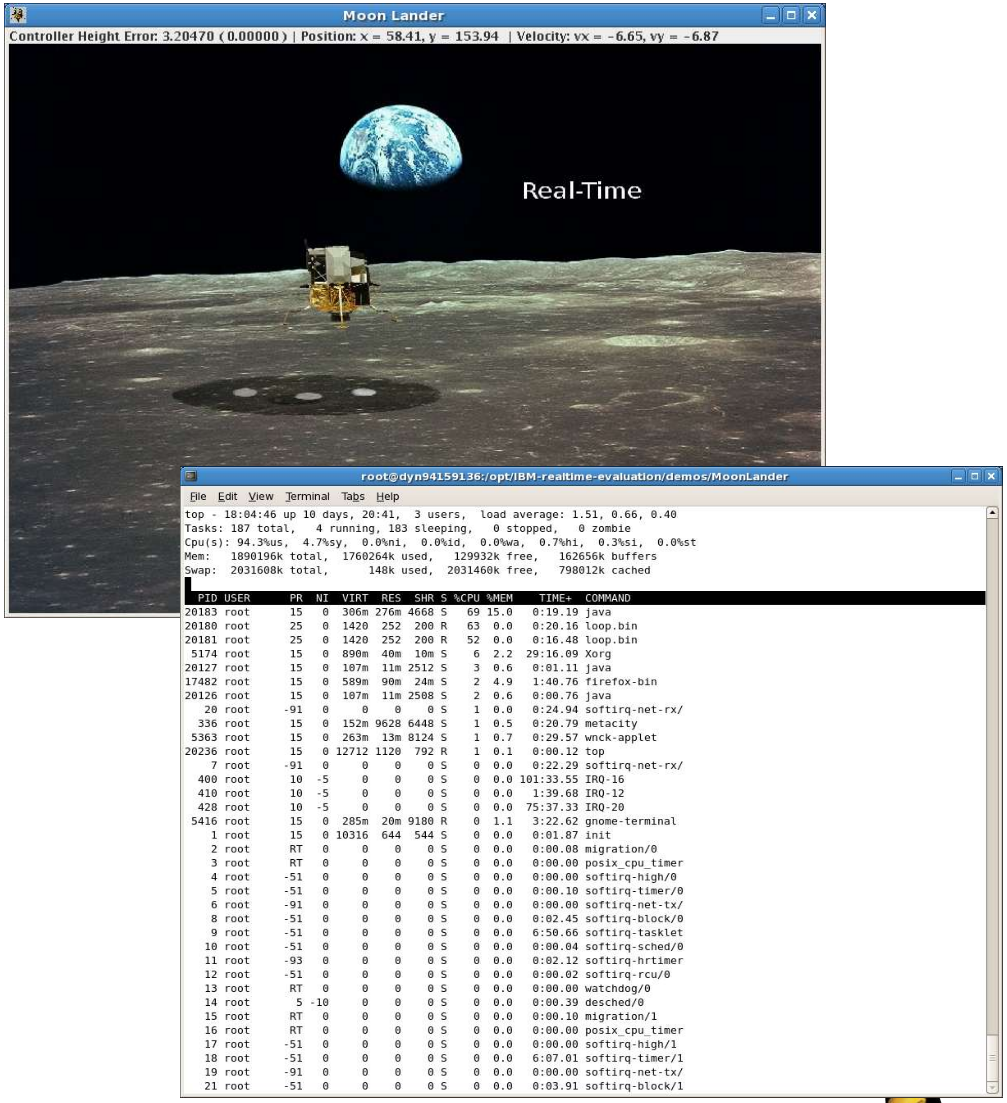
- What will I see?
  - The lunar module's movement is erratic at times. This is because the system is subject to interrupts which execute at higher priorities that can't be controlled by developers.



Real Time Linux

Moon Lander Demo (IBM Real Time)

- The scenario
  - You'll see the video of a lunar module landing.
  - In the simulation, the coordinates for the lunar module's landing area are transmitted from Earth, and the round trip time is used to estimate actual from expected trajectory.
- What will I see?
  - The lunar module moves smoothly, the calculations are happening at a higher priority than any other interrupt in the system. The developer has total control over the system priorities.



Real Time

Moon Lander Demo Results

Standard Java

IBM Real Time

Questions?