



IBM TJ Watson Research Center

The What, Why and How of Linux on BlueGene Compute Nodes

George Almási
*José
Brunheroto*
José Castaños
Gábor Dózsa
Sameer Kumar

Derek Lieber
Todd Inglett
Edi Shmueli
Albert Sidelnik

Linux on compute nodes: talk outline

- **Why bother?**
- **Evaluate, fix performance gap**
- **Supporting technology**
- **Plans for future**

Arguments for Linux on compute nodes

- **Time & effort on CNK kernel development**
 - Do we really need another O/S?
 - We need Linux anyway – runs in I/O node
- **Usability perspective**
 - Applications that will not run on CNK:
 - Java based
 - Database systems ala DB2, which require huge infrastructure
 - Distributed file systems, in-memory-databases etc
 - Applications that suffer from CNK limitations:
 - Dynamic linking, externally steered apps, loosely coupled accelerators
 - Memory limitations (no virtual memory on CNK)
 - Funny file I/O, sockets, heterogeneous applications

Why we should not bother with Linux

- **IBM comfort level with Linux commitment**
 - IP issues, open source etc.
 - Potential increase in complexity, performance loss
 - No performance compromise on a \$50 million, 2MW machine
- **Customers ambivalent about need for Linux**
 - Extensive experience with small kernels (PUMA, BG/L)
 - Most customers want can be hacked up on top of CNK
 - Python, dynamically loaded libraries
 - Multiple threads, scheduling
 - Does anyone seriously want Java on 64k processors?
 - Some customers happy to follow IBM's lead on this
 - Others intend to replace O/S with Linux regardless
- **Trend: CNK becoming more complex**

Technical arguments against Linux on comp. nodes

- **On BG/L the processors in a node are not coherent**
 - Nobody knows how to run linux on these nodes using both CPUs
- **On BG/P the network device requires fixed virtual-to-physical mapping**
 - Hard to achieve in Linux, where paging is important
 - Non-trivial fixes available (S. Kumar)
- **[Naively] measured performance on Linux is inferior to blrts on BG/L.**
 - Single-node performance
 - Scaling: O/S noise

Linux: Preliminary measurement of perf. & scaling

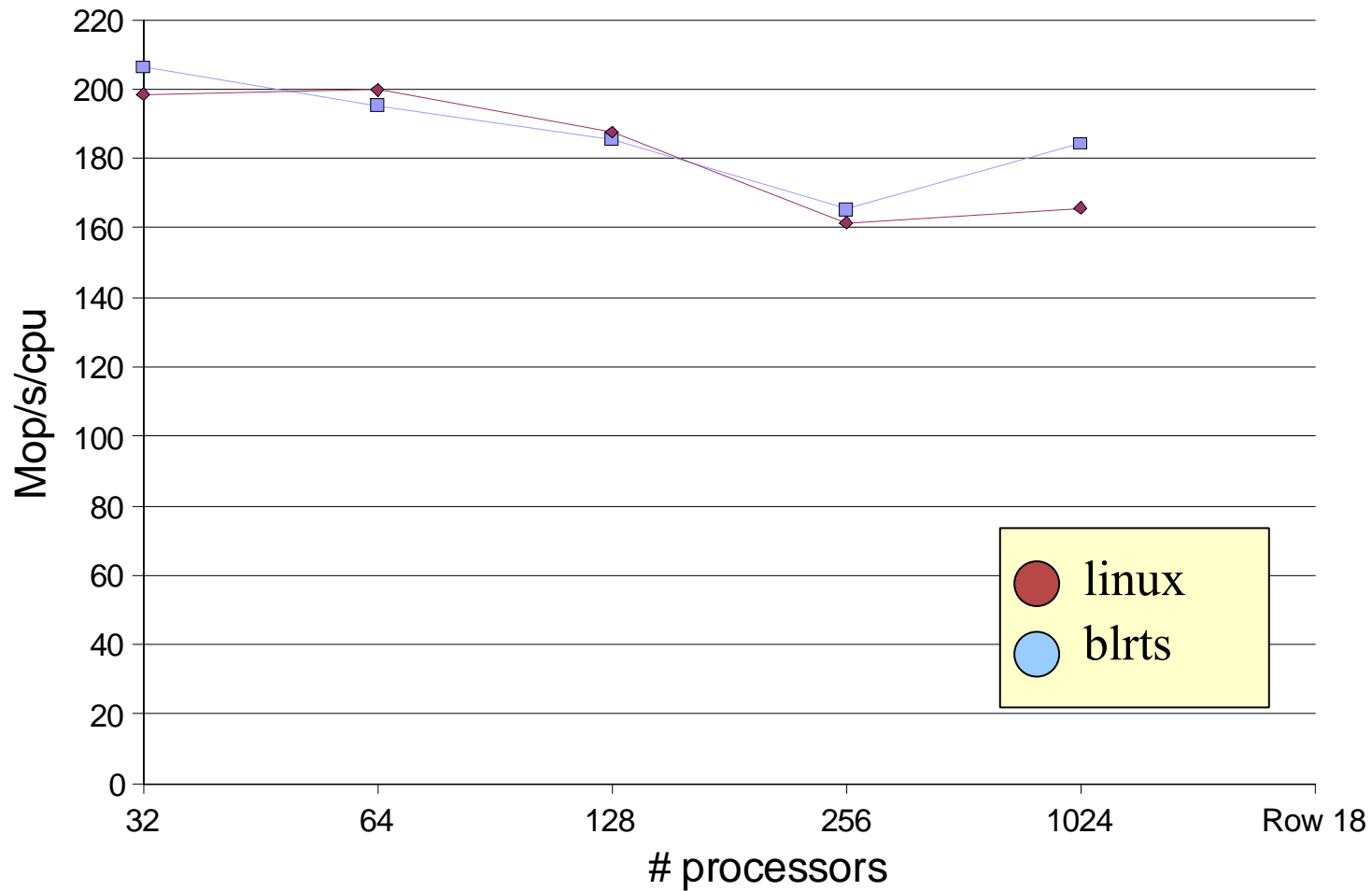
- **Goal:**

- Evaluate technical arguments objectively
- Find, eliminate sources of performance degradation
- Solve technical deployment problems

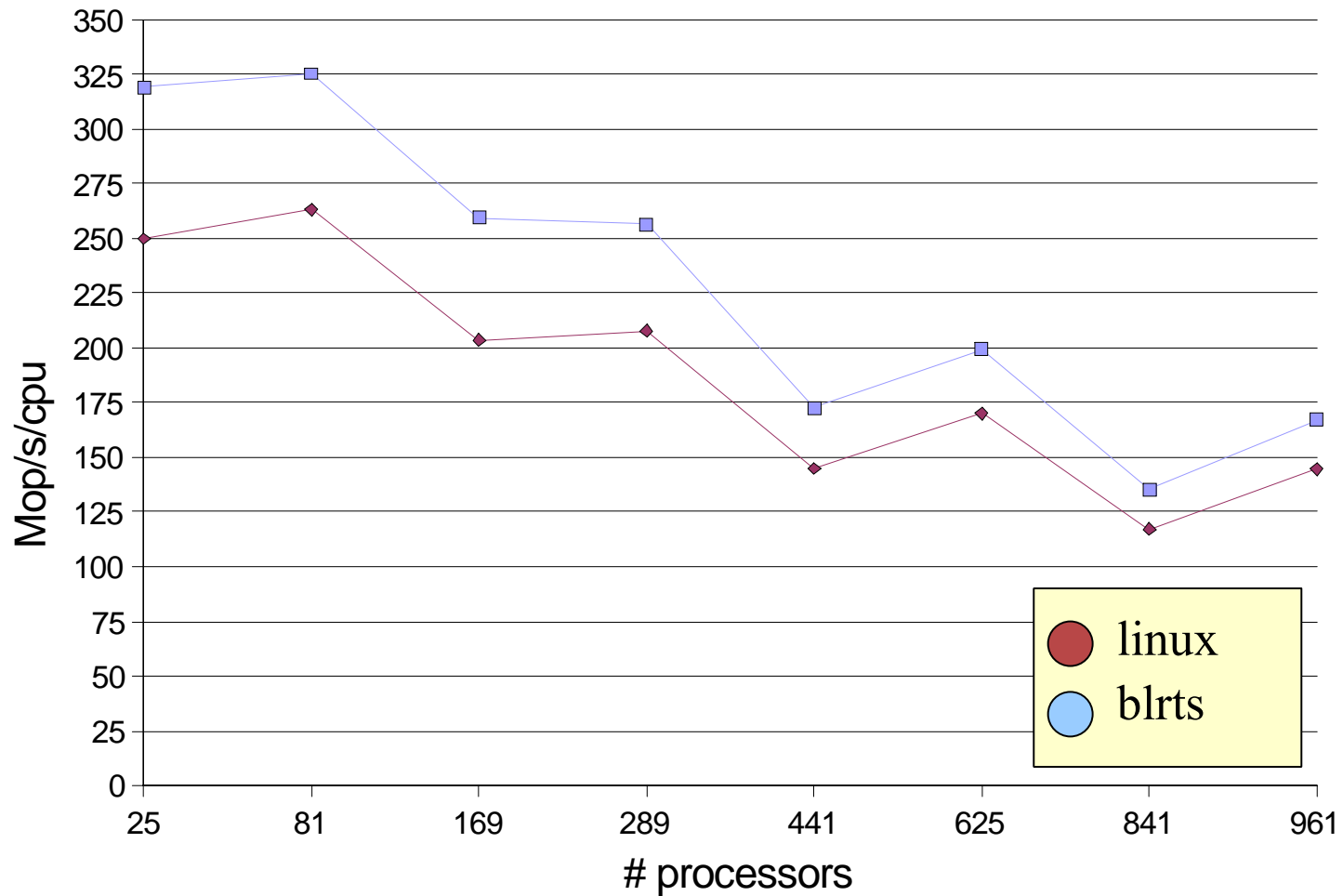
- **Methodology:**

- Used a BG/D rack for measurements
- Wrote a CIO variant for running both Linux & blrts
 - BG/L and BG/P
- Compiled & ran NAS benchmarks for blrts & Linux
- Ran microbenchmarks

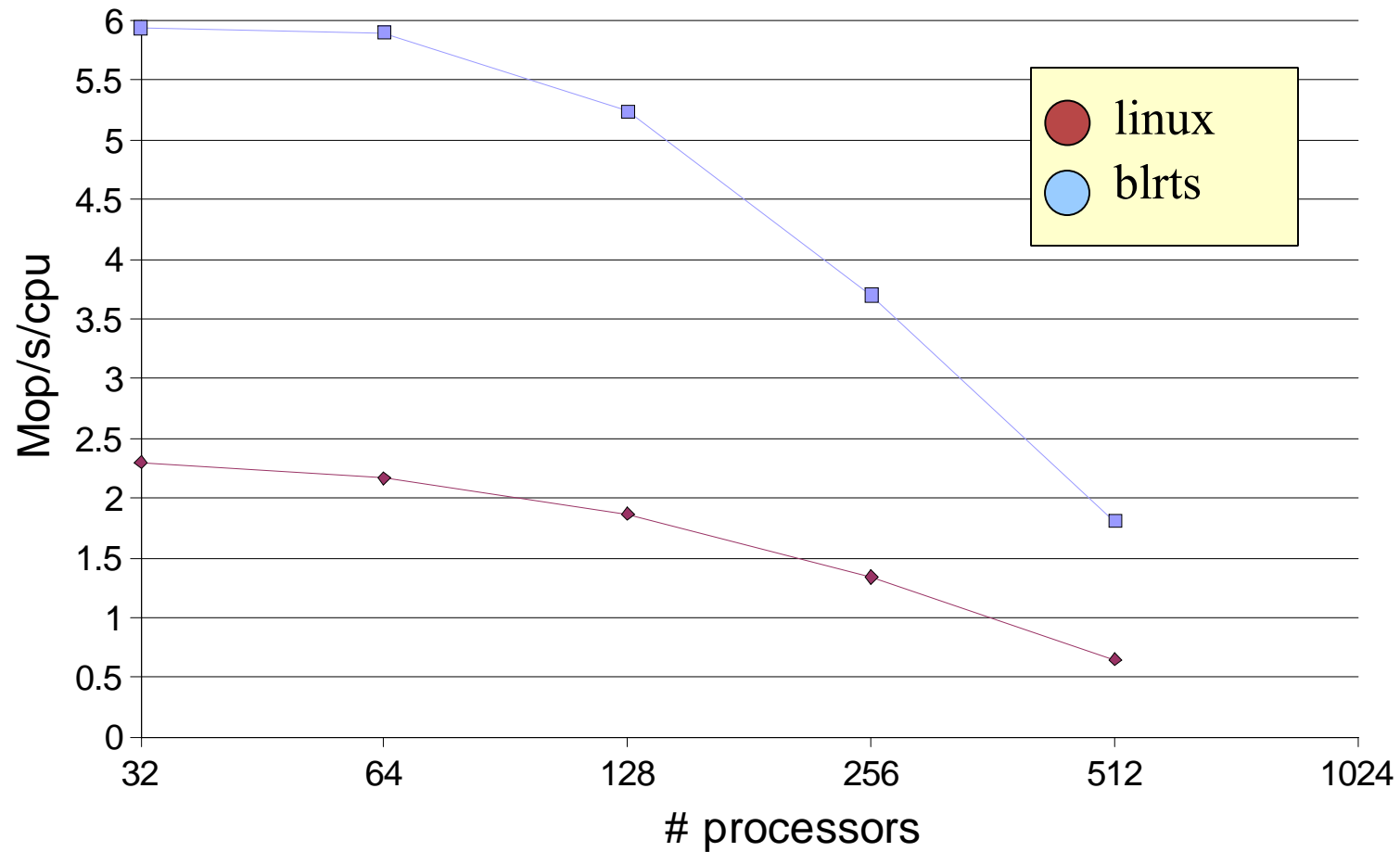
NAS Class C MG scaling: good



NAS Class C BT scaling: bad



NAS Class C IS scaling: ugly



What is wrong with NAS IS?

- **Memory access pattern:**
 - Covers > 128 Mbytes on 32 nodes
 - Randomly jumps across memory
 - On Linux, TLB coverage is $4k \cdot 62 = 252Kbytes$
 - Many, many TLB misses
 - PPC 440 handles TLB misses in software
 - On blrts (BG/L), TLB coverage is 100%
 - No TLB misses
- **How to fix it: hugetlb support for Linux**
 - Each TLB entry covers 16MB
 - 10 of 64 TLB entries cover enough for IS -> no misses

Hugetlb support in Linux: a short history

- **Early hugetlb:**
 - Allow mmap() of static hugetlb pages
 - protection from being reused by kernel
 - no paging, no COW
 - Implementations:
 - Intel, in early linux 2.6
 - R. Seth [2002]
 - 64 bit Power
 - D. Gibson [2003]
 - **No powerpc32**
- **Problem: not transparently usable**
- **Hugetlb grows up:**
 - Linux 2.6.16:
 - on-demand paging, COW
- **libhugetlb:**
 - Link application to map heap onto hugetlb
 - D. Gibson, 2006
- **Can transparently make Fortran codes use hugetlb**
- **Still no powerpc32 port**

Hugetlb on powerpc32

- **Why no powerpc32 port for hugetlb?**
 - All p-series processors today are 64 bit
 - Nobody cares about hugetlb on embedded CPUs! (we do)
- **Enter Edi Shmueli**
 - Ported hugetlb to powerpc32 (including BG)
 - Both “old” and “new” hugetlb system
 - Started the open-source process (thru LTC)
 - Examined performance impact on single-node benchmarks

Eliminating TLB Thrashing (measured by Edi)

- **Class 'A', Serial, Integer-Sort NAS Benchmark**
 - 96MB total memory footprint
 - Total of three arrays, 32MB each
 - Total of 10 iteration, each with multiple memory read/writes
 - Arrays mapping to memory:

	4KB pages	16MB pages
Number of pages	24576	6



TLB Thrashing



Eliminating TLB Thrashing (measured by Edi)

IS Benchmark Completed

Class	=	A	A
Size	=	8388608	8388608
Iterations	=	10	10
Time in seconds	=	24.44	6.51
Mop/s total	=	3.43	12.88
Operation type	=	keys ranked	keys ranked
Verification	=	SUCCESSFUL	SUCCESSFUL
Version	=	3.2	3.2
Compile date	=	31 Jul 2006	01 Aug 2006

TLB Thrashing → 24.44
→ 3.43

Linux

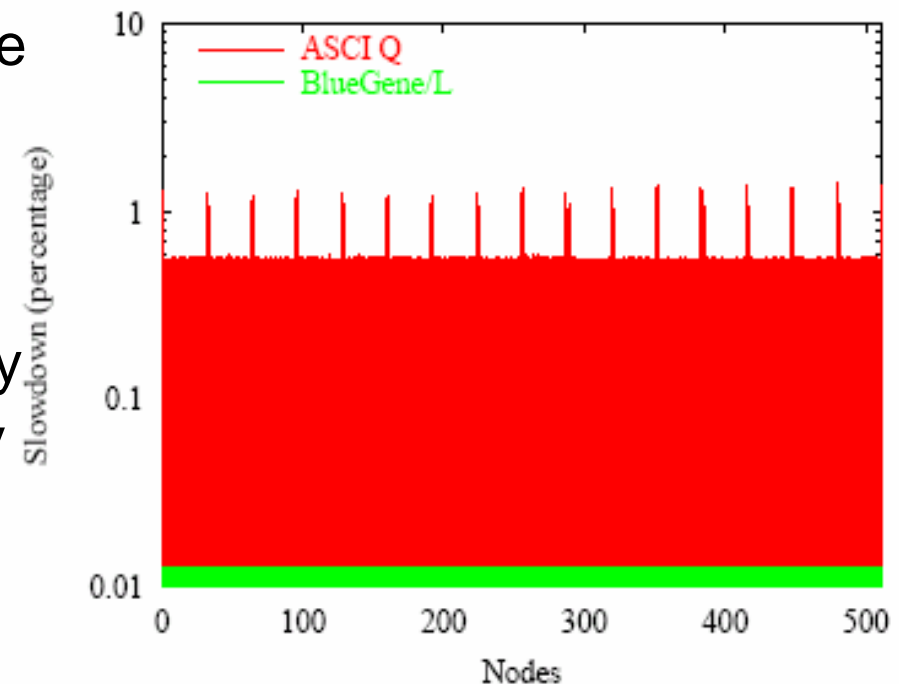
BLRTS

		A
		8388608
		10
No Thrashing	→	6.38
	→	13.15
		keys ranked
		SUCCESSFUL
		3.2
		31 Jul 2006

Linux with Huge-page support

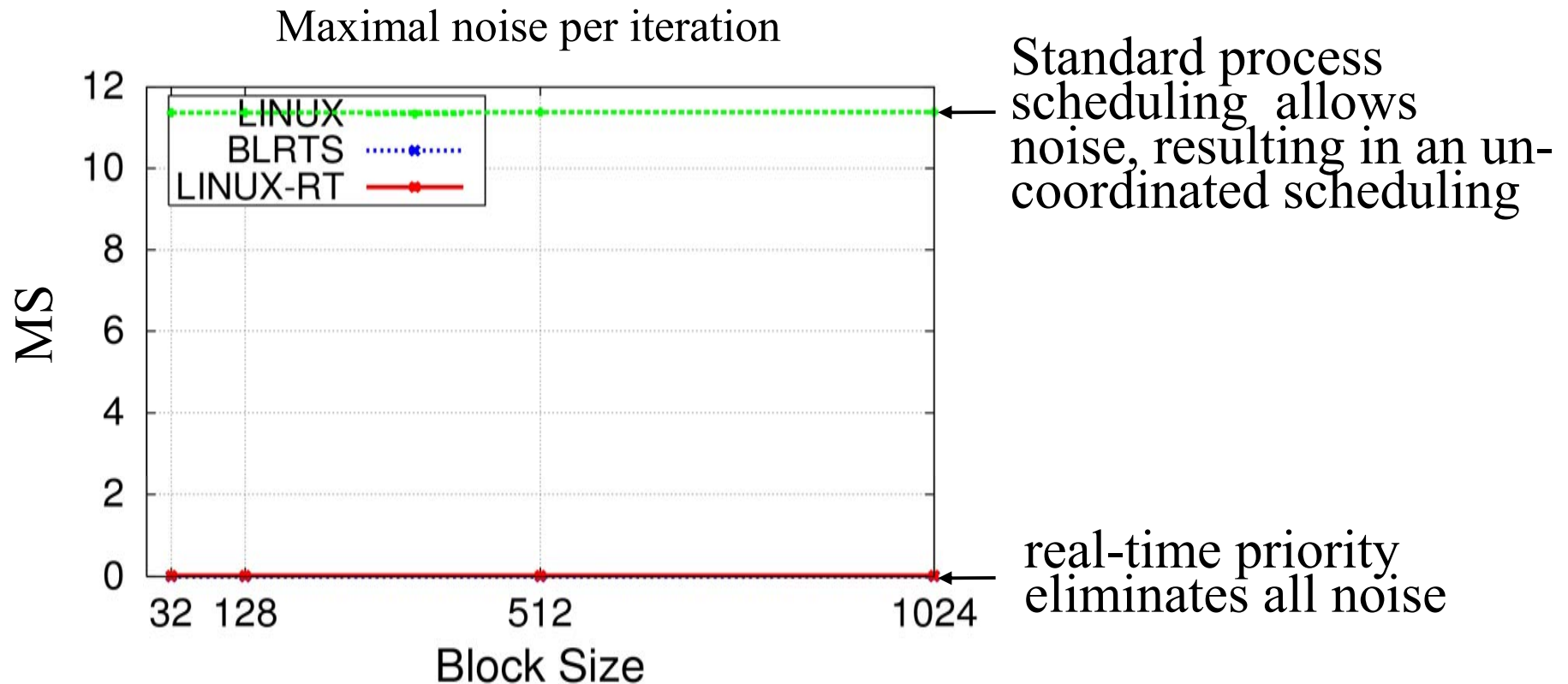
System noise on Linux

- **Scaling argument:**
 - Noisy systems don't scale
 - Linux is noisy
 - CNK is not
- **Our argument:**
 - Other machines are noisy
 - BG hardware is relatively quiet
 - We can make residual Linux noise go away



Noise comparison by LANL team [A. Hoisie] in 2003

Measuring noise on Linux (S. Kumar, E. Shmueli)



Linux on Compute Nodes: the technology

- **Boot Linux on compute nodes**
 - BG/P, BG/L, BG/L experimental control systems
 - J. Castanos, D. Lieber, A. Sidelnik
- **Train & talk to all networks**
 - port of link training to Linux “firmware” (BG/P)
 - T. Inglett, A. Tauferner, J. Brunheroto, G. Almasi
 - bglnet, bgpnet network drivers
 - E. Shmueli, G. Almasi
- **Linux distribution**
 - Special ramdisk for compute nodes (hacked by G. Almasi)
- **Job Start, function shipping of file I/O during the run**
 - Original port: D. Lieber
 - “Refined” SW architecture: G. Almasi, D. Lieber

hpcio: a “high performance” CIO protocol

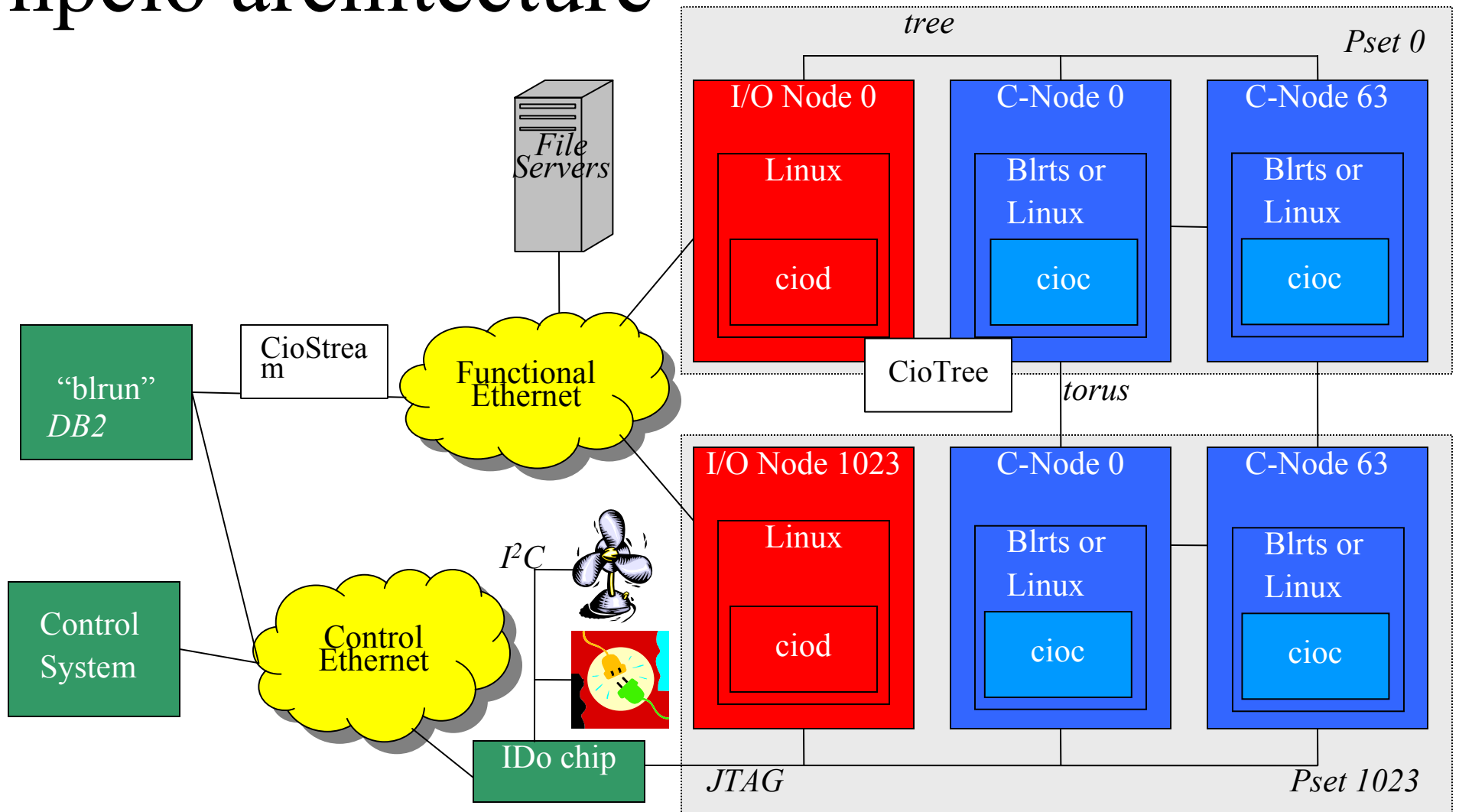
- **Same code base on /L, /P**
 - Development on /L – stable environment
 - Possible compute node targets: CNK, blrts, Linux/P, Linux/L
 - No inheritance, no virtual functions
- **Modular and extensible**
 - TCP/IP over tree implemented & tested
 - ANL to add PVFS specific functions
 - Multiple debugging tools: Etnus, Eclipse-based?
 - Heterogeneous MPI through hpcio?
- **Standardized transports**

hpcio components

- **CioTree protocol, library**
 - Transport layer for tree
 - Similar to BGML for MPI
 - G. Dozsa
- **CioStream protocol, lib**
 - Transport for CIO commands & data
 - D. Lieber, M. Mundy
- **Executables:**
 - `blrun` (job runner)
 - `ciod` (I/O daemon)
- **CIO modules, currently:**
 - Job Control
 - File I/O
 - Standard I/O
 - Tree TCP/IP

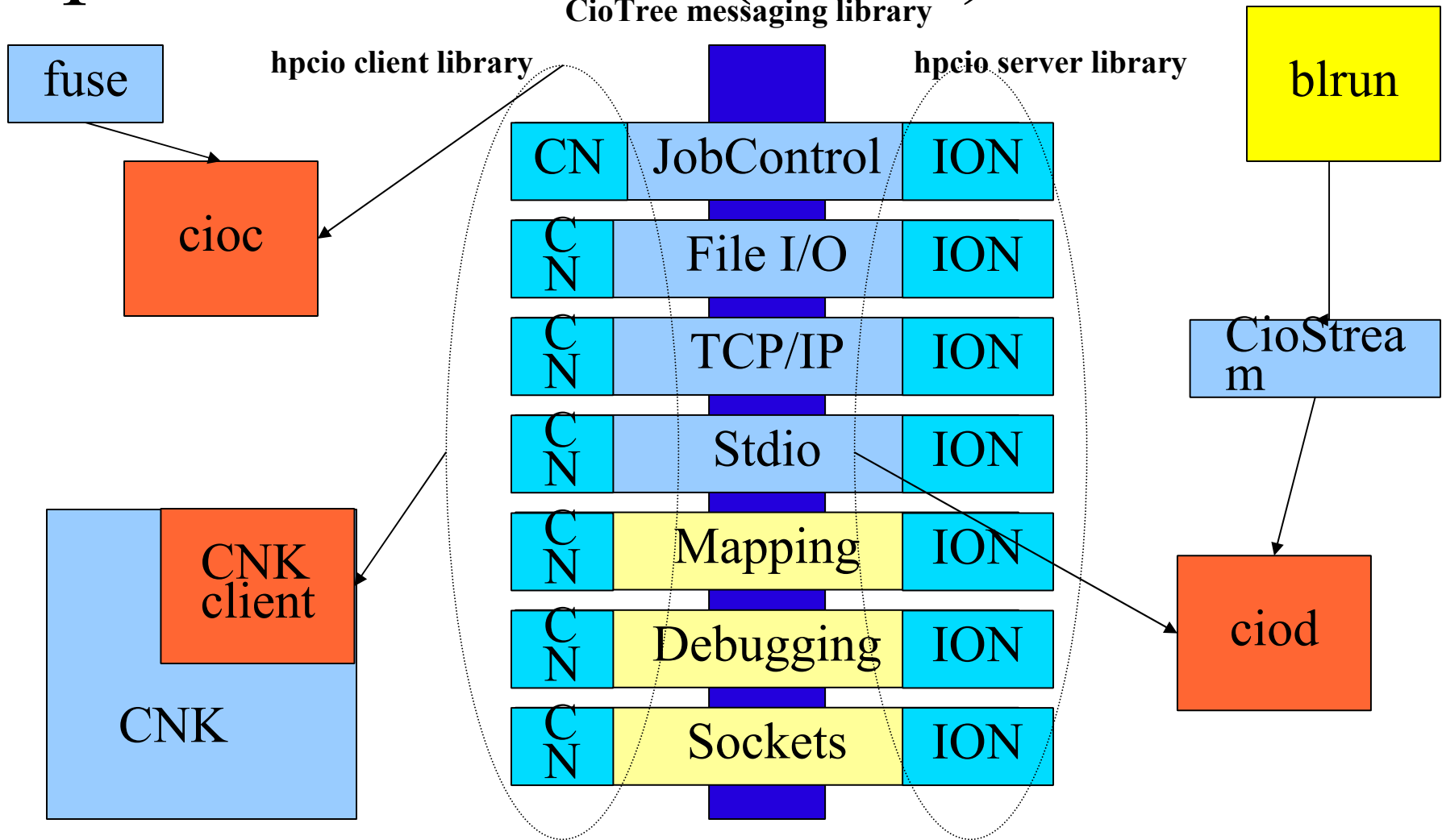
- **Linux File I/O client**
 - Uses FUSE
 - Filesystem in **USER**space
 - Miklós Szeredi (non-IBM)
- **cioc (CN daemon)**
- **CNK client library**

hpcio architecture



Hpcio architecture (detail)

CioTree messaging library



CioTree API: a tree transport (G. Dozsa)

- **portable (/X, /L, /P)**
 - “sysdep” approach pioneered in BGML
 - code shared with BGML
- **Point-to-point messages**
 - Plus ION->CN broadcast
- **Aligned buffers only**
- **No restart capability**
- **All messages have to be acknowledged**
- **Requires some infrastructure to use**
- **256 “protocols”**
- **User buffers**
 - requires $O(1)$ memory
- **Partially supports VN mode**
- **Multithreaded mode: untested**

hpcio status

- **Works on BG/L**
 - Used by Edi to run Linux tests
 - Part of the Colony project's deliverable (A. Sidelnik)
- **Port to BG/P underway**
 - Low priority; IBM Rochester wanted to port BG/L ciod first
 - Bug fixes: D. Lieber
 - Deployment T.B.D

What next?

- **Deploy hugetlb fix in Linux**
 - Measure performance
- **Deployment of hpcio underway**
 - There will be file I/O performance targets on /P:
 - maybe hpcio can help
- **Port of newest Linux kernel to /L, /P underway**
- **Other Linux applications:**
 - Distributed Java benchmarks (talking to D. Bacon)
 - TCP/IP over the torus!
 - In-memory databases w/ Linux
 - The Linux distro effort: get rid of cross-compilers!
 - Compile natively on the I/O node – or even on compute nodes

What next? (cont)

- **Continue help out with Linux firmware effort**
 - A way to address IP issues (T. Inglett)
- **Ability to run CNK/blrts and Linux at the same time**