Compute Node Linux: New Frontiers in Compute Node Operating System

Dave Wallace

Agenda

- This talk is really only about CNL Performance – and recent work in that area…
- Measures – What is interesting (or at least what is being worked on now)
- Jitter – What do we know today
- Portals – (because this is where the work has been focused)
- I/O – Baseline results
- Application Results

CNL Performance

- Measures –
  - Application Runtimes
    - Comparing QK to CNL on defined set of applications
  - I/O tests
    - I/O benchmarks
  - Application start/stop
    - Ensuring application start/stop is similar to Catamount
    - Showing benefit of no “llrd”

Analysis of applications and performance data indicate that there is little difference in single node performance – So, no issues with compiler generated code, libraries, comparable system calls – QK to CNL. Thus, the focus of the development work is on scalability issues, application communication, and I/O.

CNL Performance

- Notes –
  - This discussion is about work over the past 6 weeks and does not cover all the changes over the past 6 months.
  - The modifications described here are not complete. Our plan is to test and commit changes as they are ready. We wanted to show you what is happening in development and what progress is being made..
  - Measurements are – comparable where we can make them comparable and we explain differences where comparisons might be misleading..

CNL Performance

- Development Task Areas –
  - “Jitter” reduction –
    - Multiple approaches were possible – we chose putting Linux on a “diet” over synchronized scheduler to start
    - This work area is difficult to test and benefits are going to be less than other changes in progress
  - Portals Performance –
    - Linux Portals performance not tuned and not tuned for applications
    - Locking with multi-core needed attention
    - Memory management is different and has several known issues to pursue
    - General Portals performance differences between Catamount and CNL…

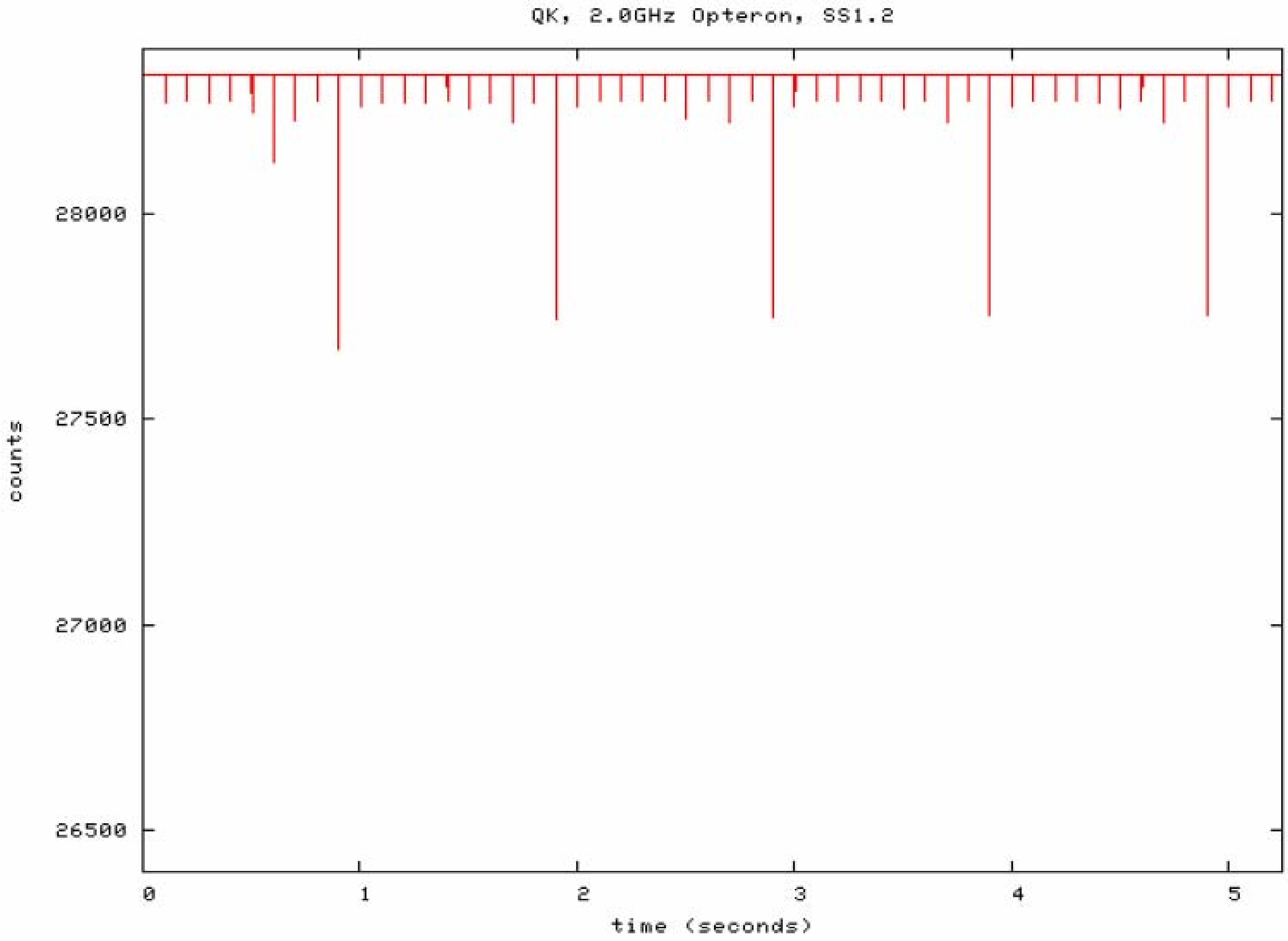
CNL Performance

- Development Task Areas –
  - I/O –
    - “Jitter” and “Spew” -
      - Lustre timeouts and console messages need work
    - Analysis of other Lustre issues is underway
  - Programming Environment
    - MPI2 –
      - Send to self changes
    - OpenMP
      - Mixed with MPI – Works now – no analysis yet…
  - Application Start/Stop –
    - Planned analysis

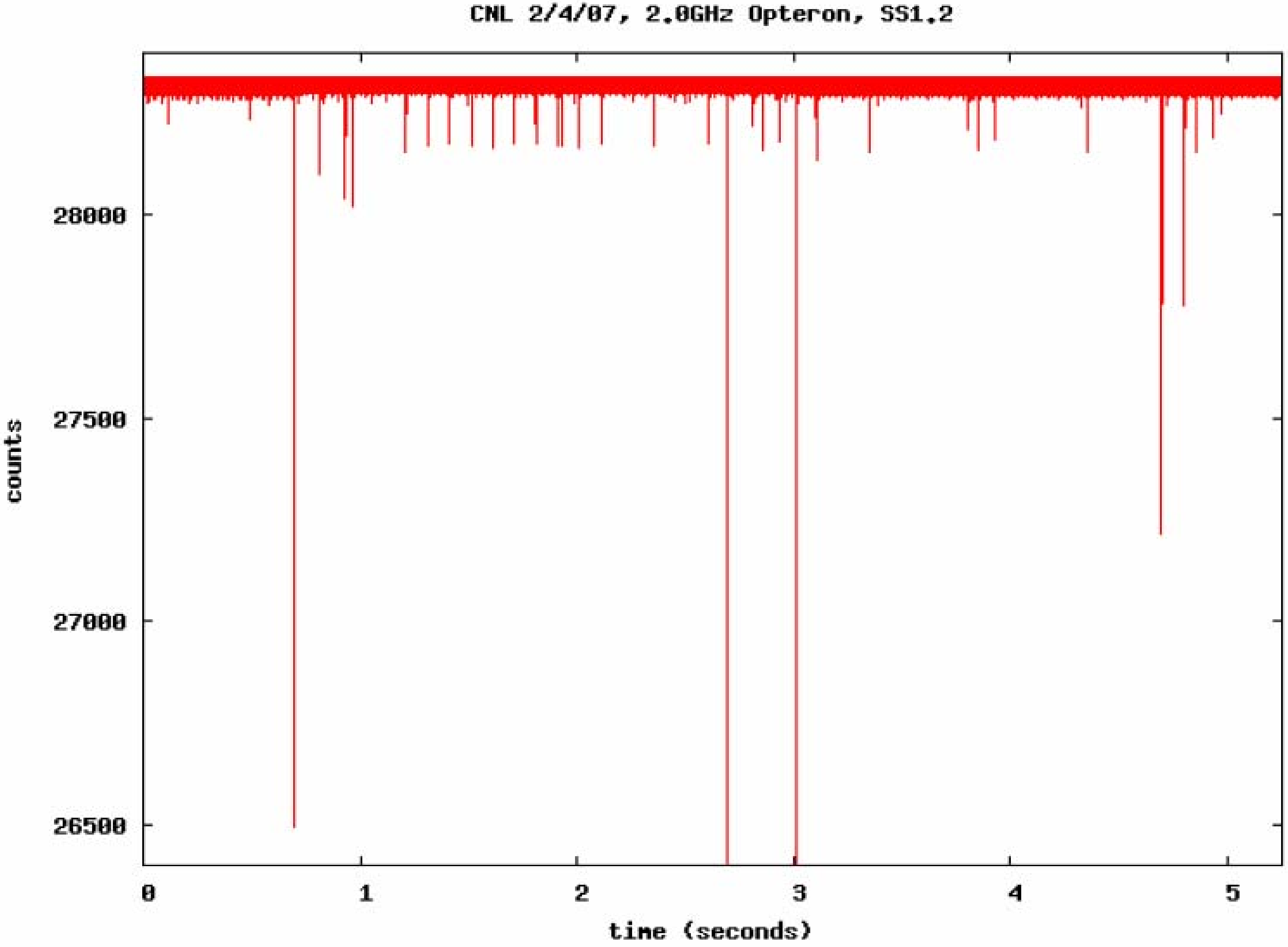
Approach being used is to not work on the applications but focus on microbenchmarks and tests that show problems we see in applications – fix the problems, integrate, and retest.

Jitter

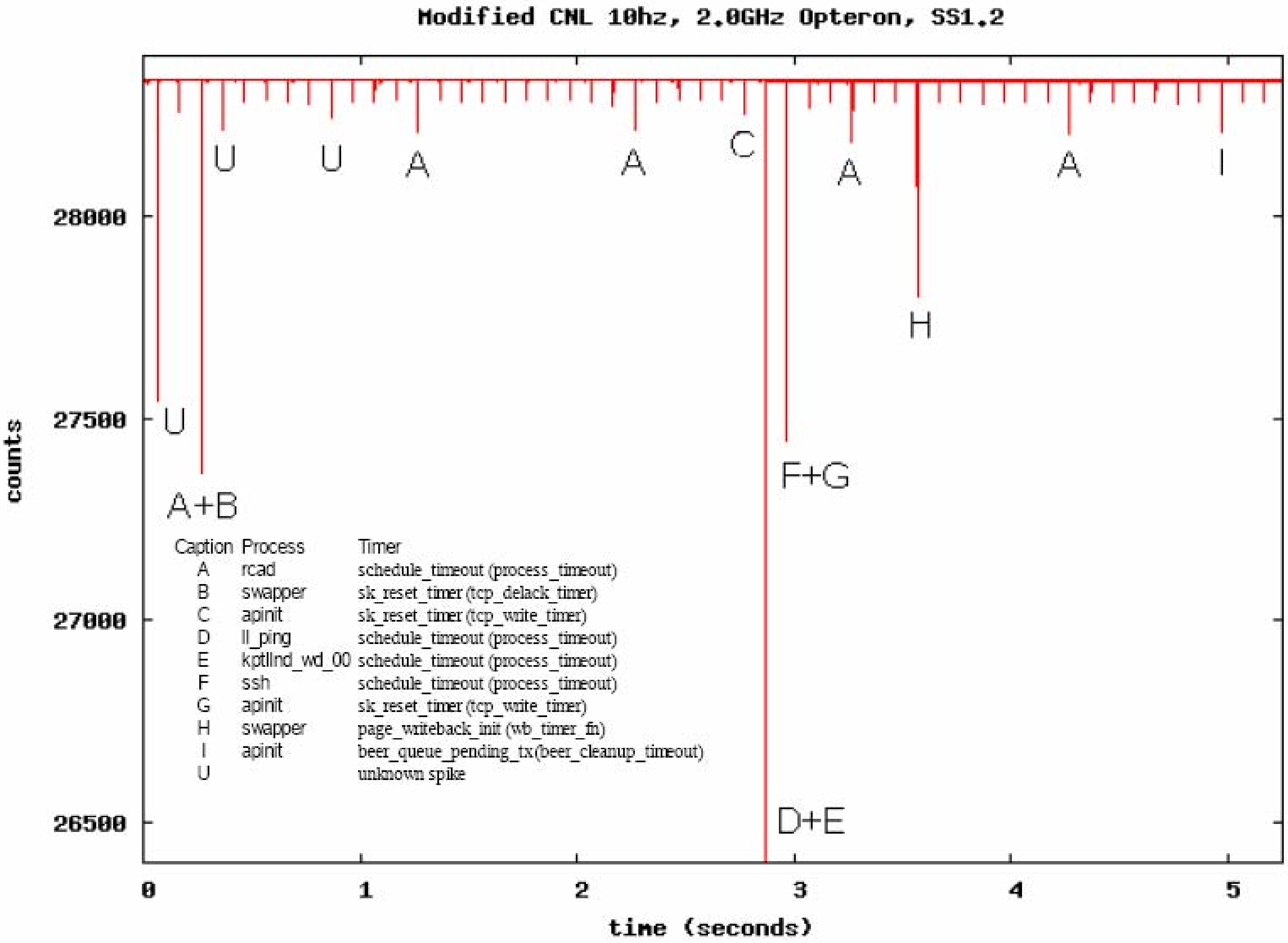
FTQ on Catamount



FTQ on Linux

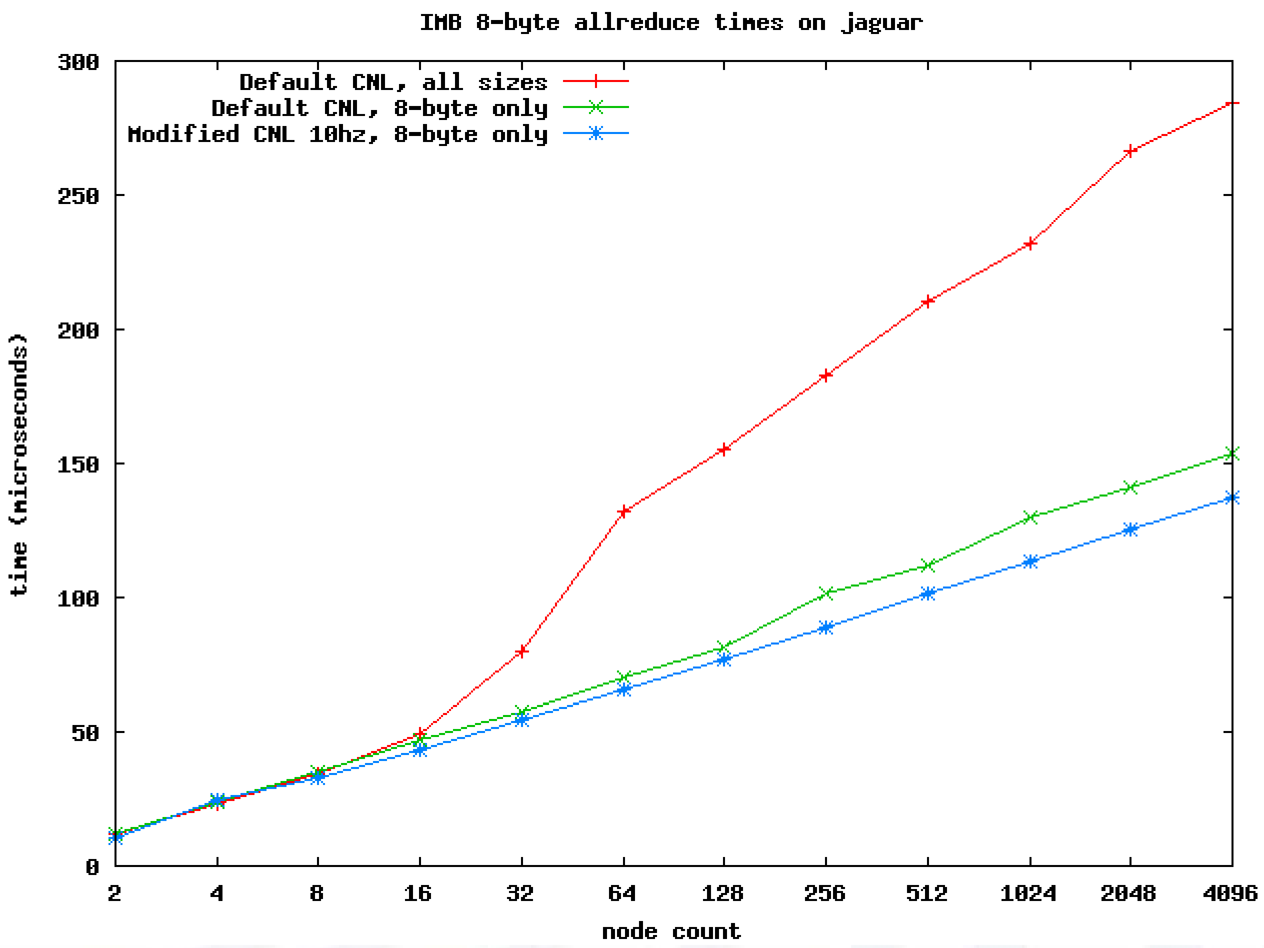

CNL 2/4/07, 2.0GHz Opteron, SS1.2
counts
28000
27500
27000
26500
0
1
2
3
4
5
time (seconds)

FTQ evolving on CNL



Caption	Process	Timer
A	rcad	schedule_timeout (process_timeout)
B	swapper	sk_reset_timer (tcp_delack_timer)
C	apinit	sk_reset_timer (tcp_write_timer)
D	ll_ping	schedule_timeout (process_timeout)
E	kptllnd_wd_00	schedule_timeout (process_timeout)
F	ssh	schedule_timeout (process_timeout)
G	apinit	sk_reset_timer (tcp_write_timer)
H	swapper	page_writeback_init (wb_timer_fn)
I	apinit	beer_queue_pending_tx(beer_cleanup_timeout)
U		unknown spike

Applied Jitter Changes - CNL

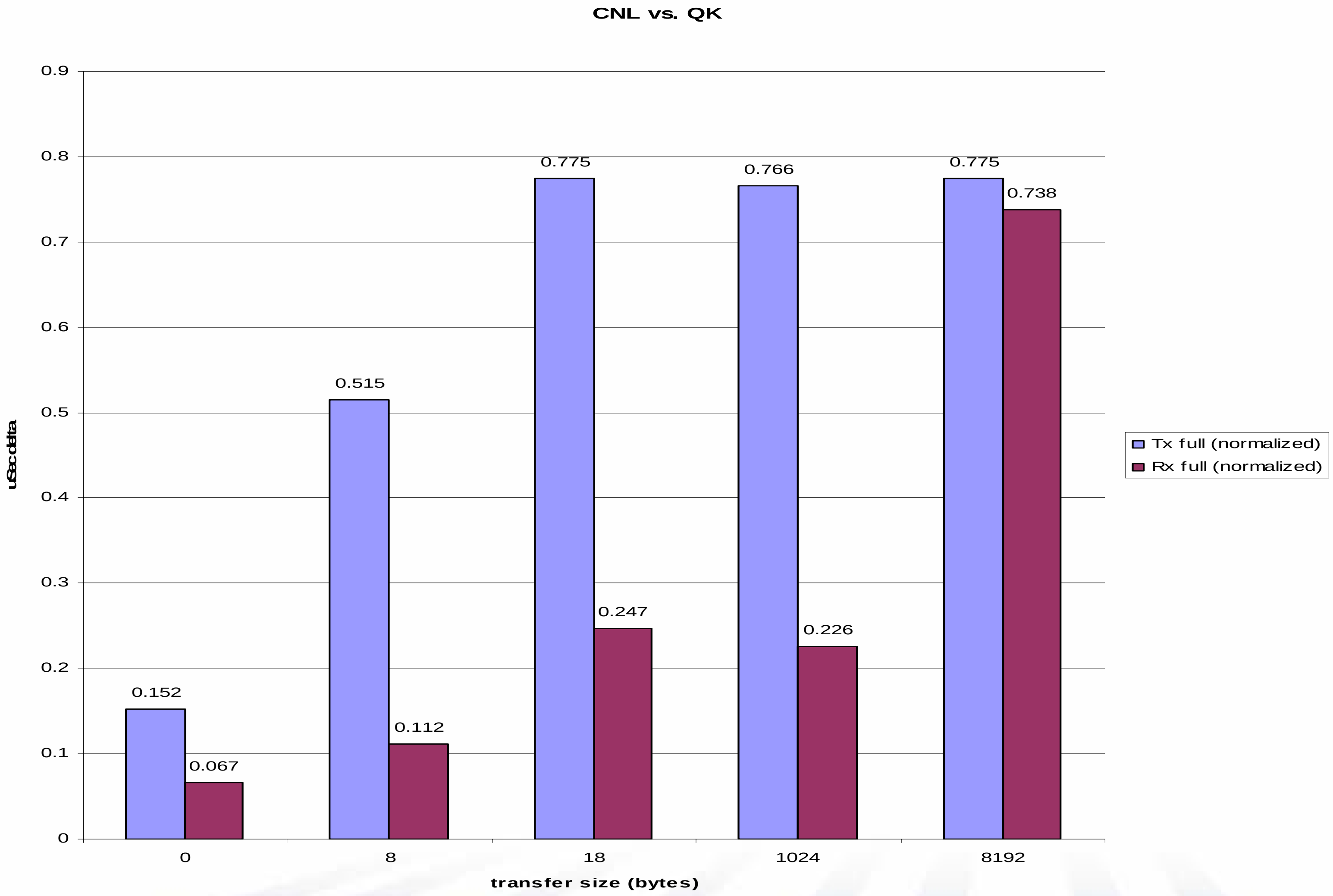


Portals

Portals Performance

- Linux Portals performance not tuned and not tuned for applications
  - Example of changes some months ago in early analysis of Portals
- Kernel locking with multi-core
  - Adding locks to make multi-core Portals more “symmetric”
- Memory management differences
  - 4K page size only in CNL, Allocation schemes differ, etc.
  - Odd behaviors in Portals that appear to be memory related
  - Reviewing memory management inside Portals – work underway now
- General Portals performance
  - 20% higher 0 byte latency
  - 15% lower bandwidth
  - Initial change to Memory management halved latency difference
  - Reviewing all the paths in drivers for differences – “Improvements come at 50-100ns at a time. The delays between ‘at a time’ increases as we approach Catamount performance levels.”

CNL Portals Overhead



I/O

I/O Performance

- Lustre expected to perform as well under CNL as QK
  - QK benefited from attention to locking and metadata management
  - CNL benefits from caching on the client
  - Some work needed to reduce Jitter in Lustre client-server heart beats

8 processor IOR single file, stripe 1 MB, count 1, start 0

Test	Size	Operation	QK	CNL	Measure	Diff %
IOR	65536	Write	119.781	183.5	MB/sec	53%
IOR	65536	Read	154.907	160.255	MB/sec	3%
IOR	1048576	Write	183.642	184.309	MB/sec	0
IOR	1048576	Read	180.608	161.95	MB/sec	-10%
IOR	4194304	Write	186.186	183.901	MB/sec	-1%
IOR	4194304	Read	186.999	175.554	MB/sec	-6%
mdtest	1	Dir_create	2328.315	3360.305	ops/sec	44%
mdtest	1	Dir_stat	4855.245	4899.317	ops/sec	1%
mdtest	1	Dir_rm	2002.235	3240.278	ops/sec	62%
mdtest	1	File_create	2770.358	3058.486	ops/sec	10%
mdtest	1	File_stat	4824.103	4939.532	ops/sec	2%
mdtest	1	File_rm	1919.210	3118.338	ops/sec	62%

8 processor IOR single file, stripe 1 MB, count 4, start 0

Test	Size	Operation	QK	CNL	Measure	Diff %
IOR	65536	Write	48.734	678.953	MB/sec	1293%
IOR	65536	Read	123.220	569.569	MB/sec	362%
IOR	1048576	Write	524.558	688.652	MB/sec	31%
IOR	1048576	Read	404.950	517.401	MB/sec	28%
IOR	4194304	Write	725.626	693.292	MB/sec	-4%
IOR	4194304	Read	603.823	537.106	MB/sec	-11%
mdtest	1	Dir_create	2402.801	3484.987	ops/sec	45%
mdtest	1	Dir_stat	4854.606	4887.840	ops/sec	1%
mdtest	1	Dir_rm	2000.467	3161.859	ops/sec	58%
mdtest	1	File_create	1514.006	2716.896	ops/sec	79%
mdtest	1	File_stat	4686.043	4740.014	ops/sec	1%
mdtest	1	File_rm	1658.783	2685.556	ops/sec	62%

Current Application Results

Application Performance (Apr 17, 07)

Application	# Processes	% difference SC	% difference DC
GTC	512	-2	-2
	1024	-2	-2
	2048		+1
MILC	512	-10	-4
	1024	-10	-5
	2048		-4

*** GTC *** With -small_pages Weak Scaling.
shark 2.8 GHz pgi 6.2.5 QK 1.5.42 April 07, 2007
CNL 2.0-dev+ April 17, 2007

*** MILC *** Weak scaling test case.
shark 2.8 GHz pgi 6.2.5 QK 1.5.42 April 07, 2007
CNL 2.0-dev+ April 17, 2007

Application Performance (May 6, 07)

Application	# Processes	% difference SC	% difference DC
POP Step/Total	1000	-3	-9
Baroclinic	1000	0	-13
Barotropic	1000	-7	-2
POP Step/Total	2000	-10	-7
Baroclinic	2000	-1	-15
Barotropic	2000	-16	-3
POP Step/Total	4800	-1	-14
Baroclinic	4800	-4	-10
Barotropic	4800	0	-9
POP Step/Total	8000		-13
Baroclinic	8000		-12
Barotropic	8000		-13

Application Performance (May 6, 07)

Application	# Processes	% difference SC	% difference DC
POP Step/Total	10000		-14
Baroclinic	10000		-16
Barotropic	10000		-14

*** POP *** Time in seconds of 1 nday steps with .1 degree test case.
jaguar 2.6 GHz
QK 1.15.25 November 08, 2006
pgi 6.1.6 CNL 2.0.03+ May 06, 2007

Application Performance (May 6, 07)

Application	# Processes	% difference SC	% difference DC
LSMS bcc_Fe_1024	1024	-4	-4
bcc_Fe_2048	2048		-2
bcc_Fe_4096	4096		-2
bcc_Fe_8192	8192	-1	-1

*** LSMS *** LSMS 2.0i jaguar 2.6 GHz
pgi 6.2.5 QK 1.5.31 April 10, 2007
pgi 6.1.6 QK 1.5.31 April 27, 2007
pgi 6.1.6 CNL 2.0.03+ May 06, 2007

Application Performance (May 6, 07)

Application	# Processes	% difference SC	% difference DC
S3D	1024	0	+6
	2048	+13	-4
	4096	-2	+11
	8192		X

*** S3D ***
50 Time steps for these shorter runs. iobuf with QK, IOBUF_PARAMS='*'
jaguar 2.6 GHz pgi/6.1.6 QK 1.5.31 April 27, 2007
pgi 6.1.6 CNL 2.0.03+ May 06, 2007

Questions?