Reducing Application Runtime Variability on Jaguar XT5

Presented by
Kenneth D. Matney, Sr.

Sarp Oral, Feiyi Wang, David A. Dillow,
Ross Miller, Galen M. Shipman, Don Maxwell,
Dave Henseler, Jeff Becklehimer, Jeff Larkin

Operating system (OS) noise

- Interference generated by OS preventing compute core from performing *useful* work
  - Kernel daemons, network interfaces, other OS services
  - Vary in duration and frequency
- Cause de-synchronization (jitter) in collective communications
  - Variable (degraded) overall parallel application performance
  - In a tree based collective OS noise may be propagated up the tree with each node contributing system noise according to a probability distribution
    - *MPI_Allreduce*

Operating system (OS) noise

- OS noise can impact performance of tightly coupled operations
- Probability of hitting larger magnitude OS noise events increases as nprocs grows
- Large-scale applications using certain types of collective communication primitives are more susceptible

OS Noise on Cray XT5

- Varying and degraded application performance at scale
  - Observed on Jaguar XT5
  - Parallel Ocean Program (POP)
    - Heavily uses MPI_Allreduce
- OLCF and Cray investigated the problem
  - Identified major compute node OS noise sources
  - Developed a prototype Reduced Noise kernel
    - Based on UNICOS 2.2

Prototype Reduced Noise kernel

Major OS noise sources

- Kernel level noise sources
  - TCP/IP protocol
  - Time-of-Day clock
  - Kernel work queues
  - Non-fatal machine checks
  - Page cache flushing
  - DVS protocol
  - Lustre protocol
  - BEER threads
  - Virtual-to-physical memory mapping
  - Other generic timer events
- User level noise sources
  - ALPS daemon
  - RCA
    - Heartbeat, console
  - SSH
  - NTP

Solution

- Aggregate and merge OS noise sources onto a single compute core for each node
  - Cray CLE prototype kernel (based on stock 2.2 kernel)
  - Core 0 reserved for overhead only
  - Lustre/DVS processing and mapping of incoming packets are not merged
    - Application generated, not OS noise

Solution

- Exclude the “overhead core” and run scientific applications on remaining cores per node
  - *aprun -N 7 -cc 1-7 <binary>*
  - *aprun* -n 1024 *-N 8* → *aprun* -n 896 *-N 7 -cc 1-7*
- Not new but proven method, used on Intel Paragon in ’90s

Testbed

- Proof of the concept tests
  - Chester (OLCF quad core XT5)
    - Single cabinet, 60 node, 480 cores in total
- Large-scale tests
  - Jaguar (OLCF quad core XT5)
    - 220 cabinet, 18,000 nodes, 144,000 cores in total (at the time of testing)
  - Shark (Cray quad core XT5)
    - 12 cabinet, 1,065 nodes, 8,520 cores in total

Proof of the concept tests

- FWQ benchmark
  - Fixed work quanta
  - Measure how long it takes to perform a fixed amount of work
  - Report consumed cycles for every work quanta
  - Major deviations between quanta are indications of OS Noise
- Kurtosis
  - Can be used to summarize and analyze deviations

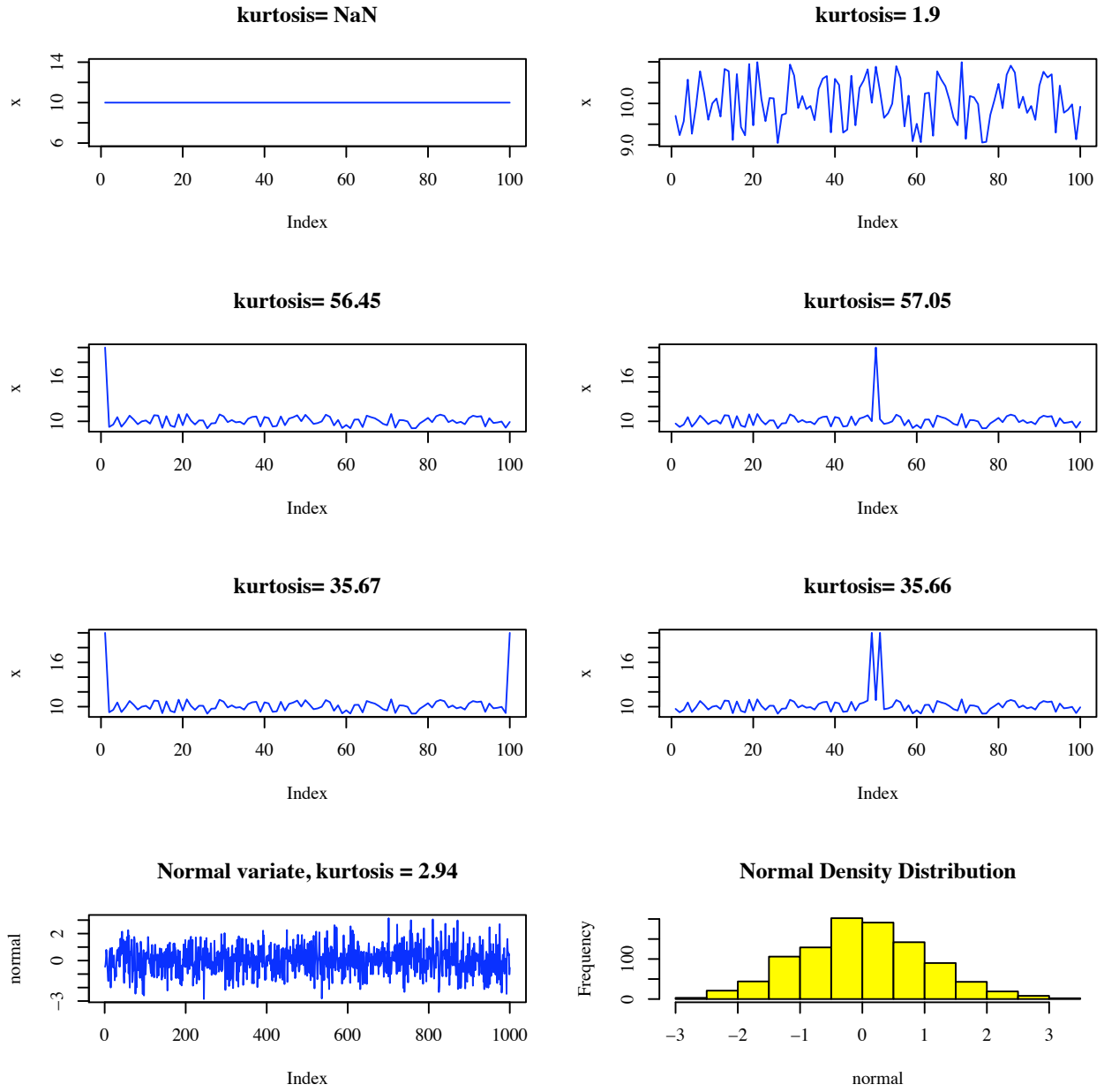
Proof of the concept tests - Kurtosis

- Kurtosis is the 4th standardized moment

$$\frac{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^4}{\left(n-1\right)\times s^4}=\frac{\mu_4}{\sigma^4}$$

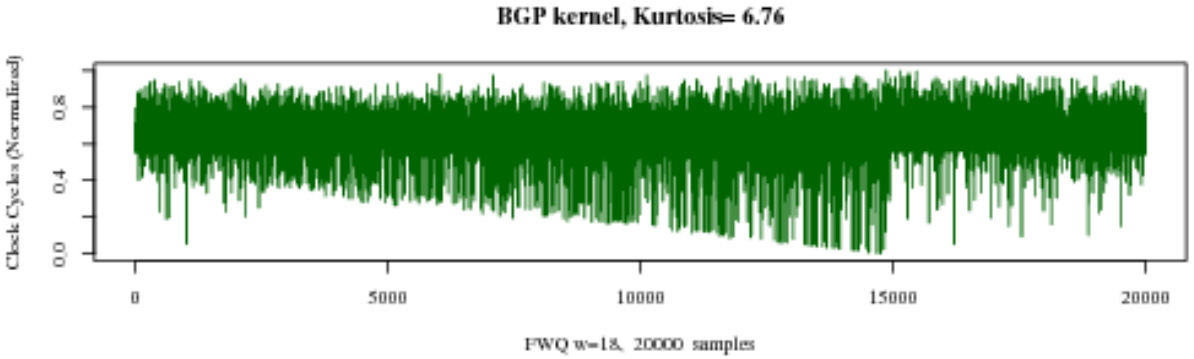
- A high kurtosis has sharp peaks and long fatter tails; a low kurtosis has more rounded peaks and short thinner tails
- Kurtosis is a common metric in noise benchmarking, but it should not be used as a sole descriptor

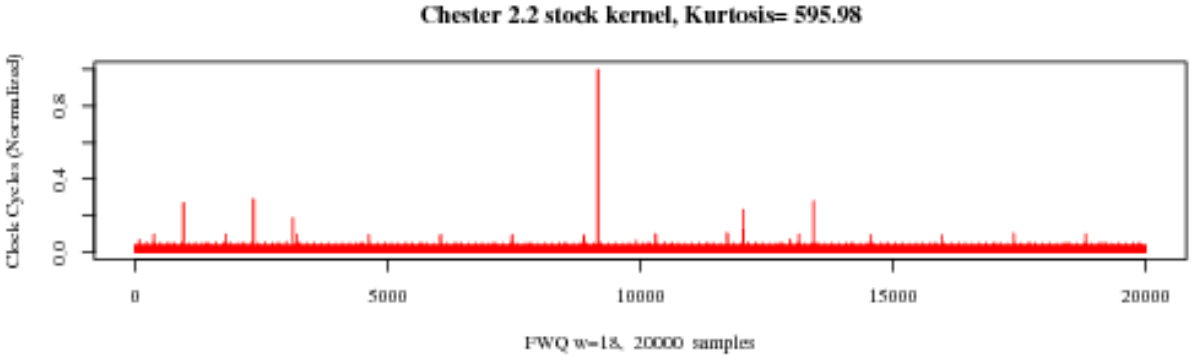
Proof of the concept tests - Kurtosis

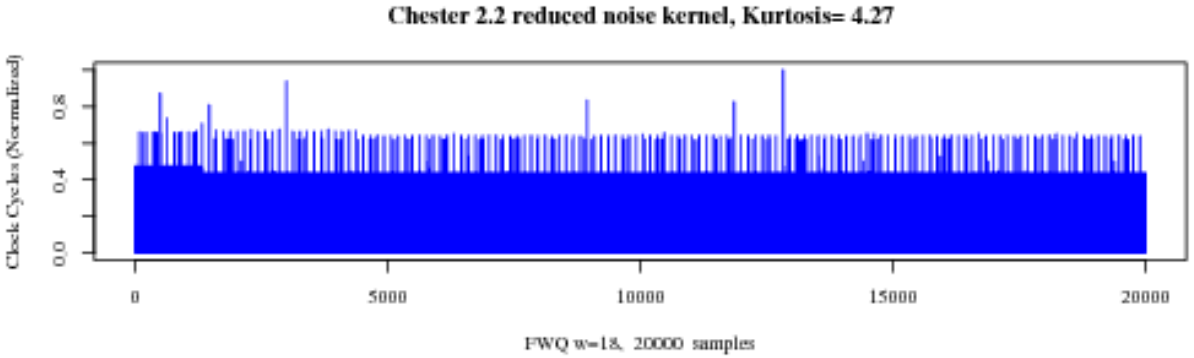


Proof of the concept tests

- Kurtosis calculated based on FWQ data
  - IBM BG/P
    - 6.76
  - Chester w/ stock kernel
    - 595.98
  - Chester w/ RN kernel
    - 4.27

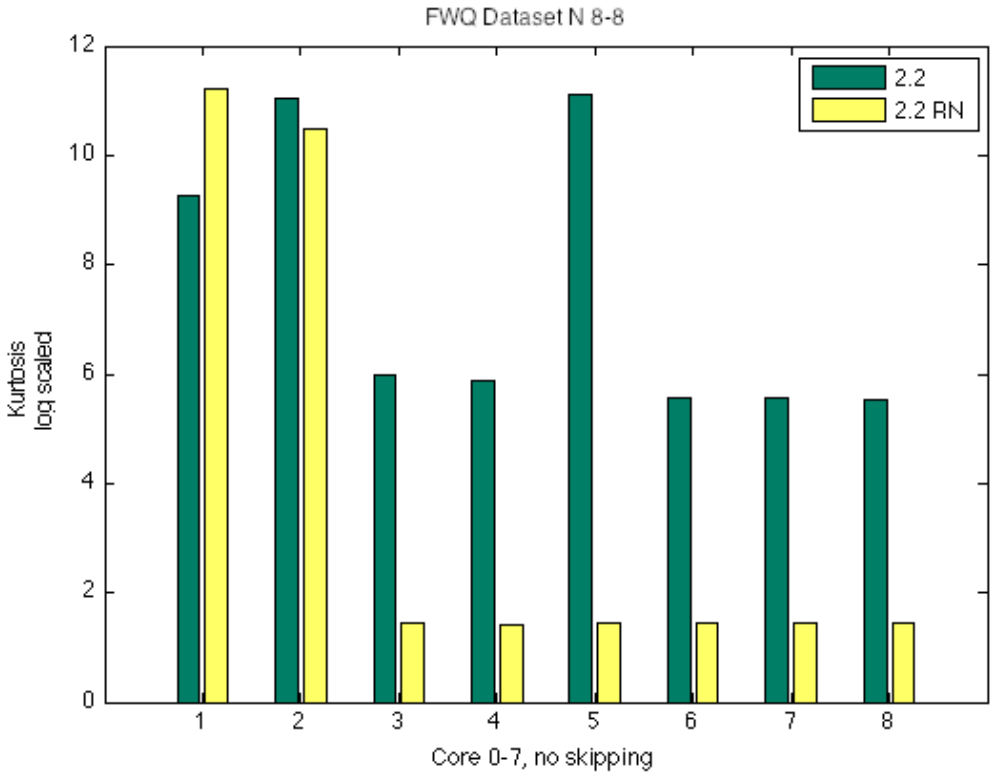






Proof of the concept tests – per core noise

- Per core noise levels
  - w/ 2.2 stock kernel
  - w/ 2.2 RN kernel
  - FWQ benchmark (threaded)
- Reduced Noise kernel
  - Substantially suppressed noise on cores 2-6
    - Uniform low noise
  - Core 0 and 1 had 4 orders of magnitude higher kurtosis

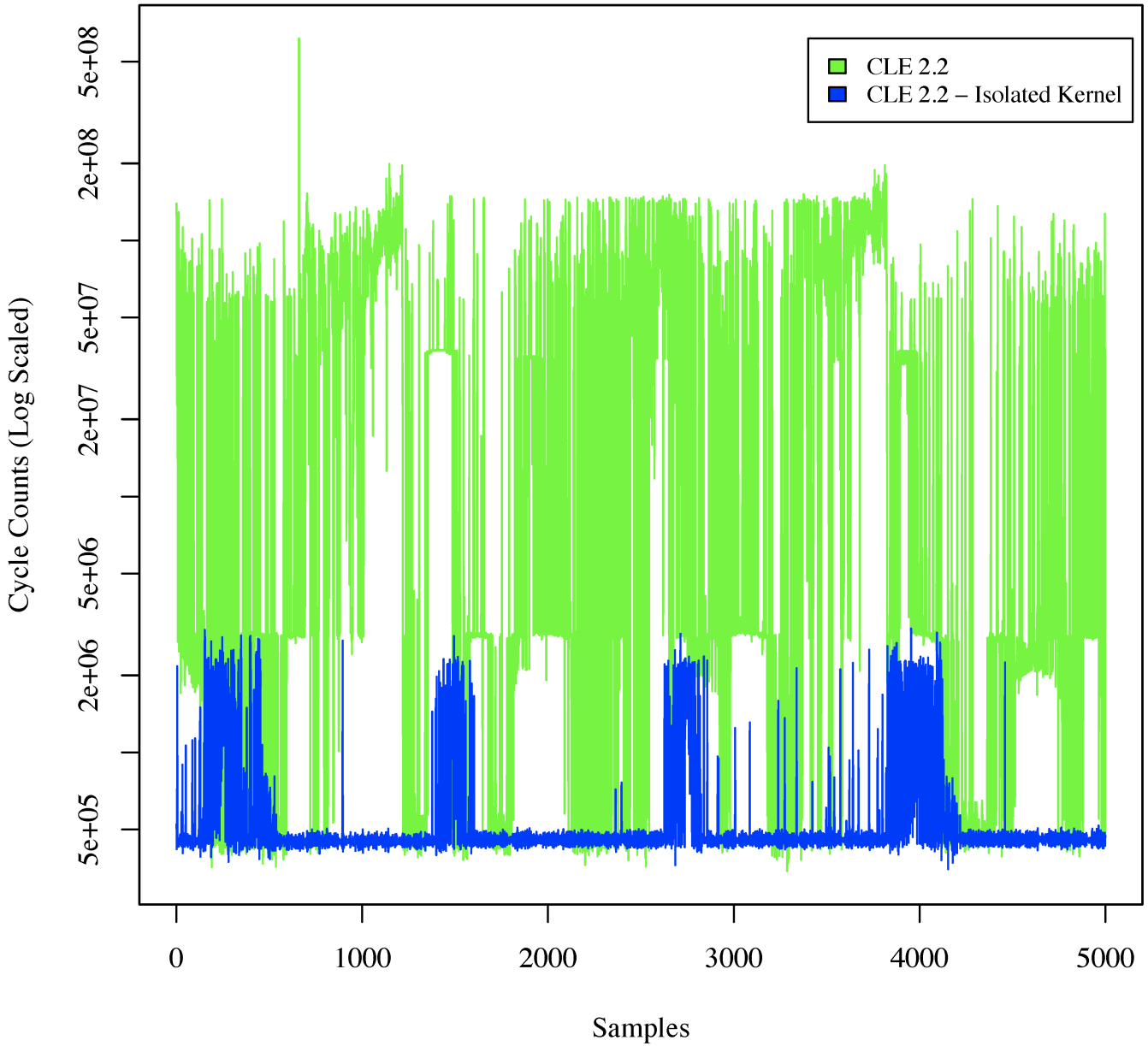


At scale tests – MPI-FWQ

- On Jaguar XT5 using 49,152 cores
- MPI-FWQ
  - In house benchmark
    - Work (w=18) + MPI_Allreduce
    - Message size = 1 MB
    - Rank 0 was root
- Excluded cores 0 and 1
  - *-N 6 –cc 2-7*

- 2 orders of magnitude improvement in MPI_Allreduce at scale

At scale tests – MPI-FWQ



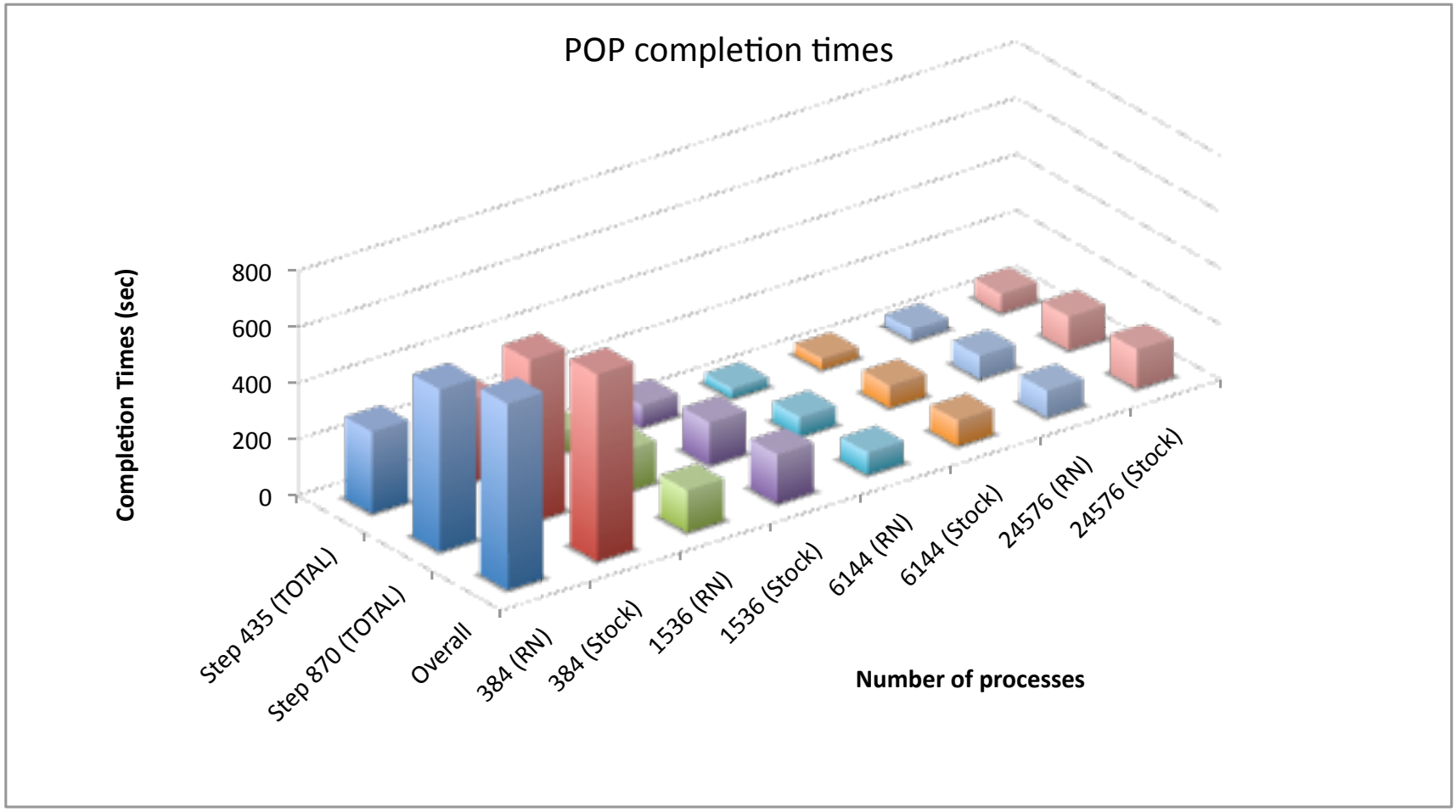
At scale tests – Parallel Ocean Program (POP)

- POP was run on Jaguar XT5 (OLCF) up to 24,576 cores
  - 2.2 Stock kernel vs. 2.2 Reduced Noise kernel
  - *-N 6 -cc 2-7*
    - Same node and core count for both kernels
  - Strong scaling
  - 1,000 steps in total
  - I/O was disabled
    - History, movie, tavg, and xdisply were all disabled
  - POP completion times measured (in seconds)

At scale tests – Parallel Ocean Program (POP)

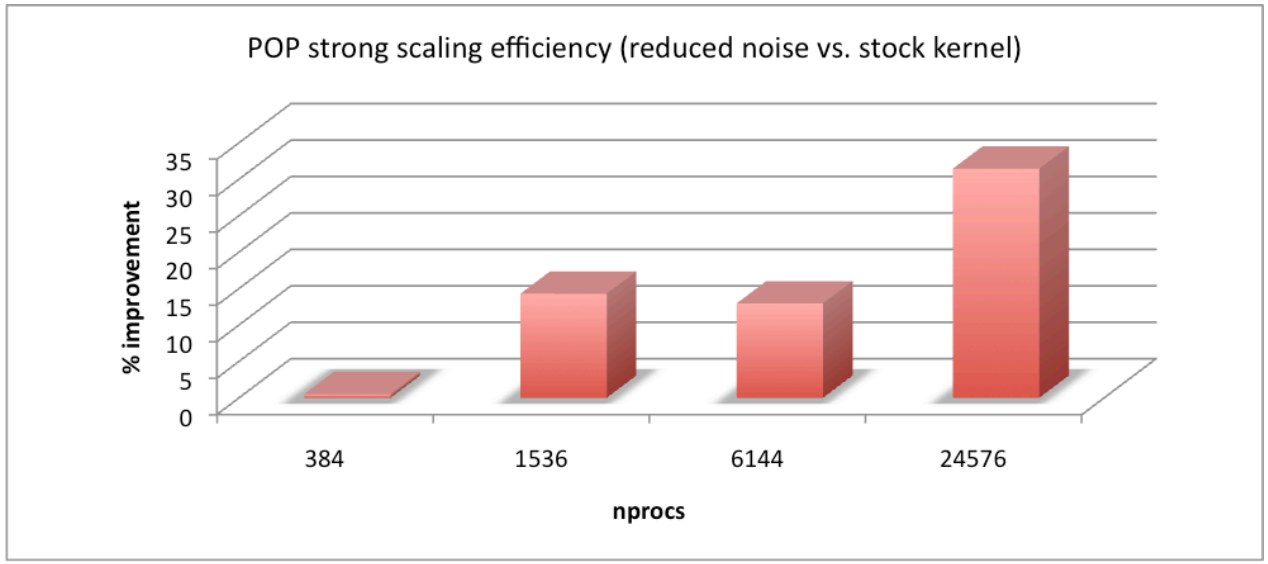
Number of Processes	Reduced Noise kernel			Stock kernel		
	Step 435	Step 870	Step 1,000	Step 435	Step 870	Step 1,000
384	289.68	575.48	660.03	291	578.09	663.13
1,536	75.27	149.16	149.16	77.46	151.94	173.98
6,144	35.33	69.17	79.13	39.17	79.25	90.89
24,576	42.7	81.78	94.58	68.43	122.79	137.94

At scale tests – Parallel Ocean Program (POP)



At scale tests – Parallel Ocean Program (POP)

- For all core counts Reduced Noise kernel performed better compared to Stock noise kernel
  - ~30% gain at 24,576 cores



At scale tests – Parallel Ocean Program (POP)

- POP was run on Shark XT5 (Cray)
  - 8,192 cores with Stock kernel
    - *-N 8*
  - 7,168 cores with Reduced Noise kernel
    - *-N 7 –cc 1-7*
  - Same node count (1,024) for both kernels
  - 2,000 POP steps in total
  - I/O disabled
- ~ 30% performance improvement with less number of cores with Reduced Noise kernel

	Number of Processes	Step 2,000
Reduced Noise	7,168	379.03
Stock	8,192	499.00

Conclusions

- OS noise is a key limiting factor on large-scale tightly-coupled applications
  - Jitter (synchronization) problem
  - More observable with some MPI collectives
    - MPI_Allreduce
- Cray CLE UNICOS 2.2 prototype kernel
  - Core 0 is
    - User selectable (per job)
    - Designated overhead core

Conclusions

- Prototype Reduced Noise kernel
  - Uniform and less noisy cores (cores 2-7)
    - In production RN kernel, core 1's noise problem is fixed
- 2 orders of magnitude improvement in MPI_Allreduce performance at scale
- 30% performance improvement in POP completion time at scale

Questions?

Contact

Galen Shipman (gshipman@ornl.gov)

Thank you!