Exploring Memory Management Strategies in Catamount

Kurt Ferreira, Kevin Pedretti, and Ron Brightwell

Scalable System Software Group

Sandia National Laboratories

Cray Users Group

Helsinki, Finland

May 8, 2008

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

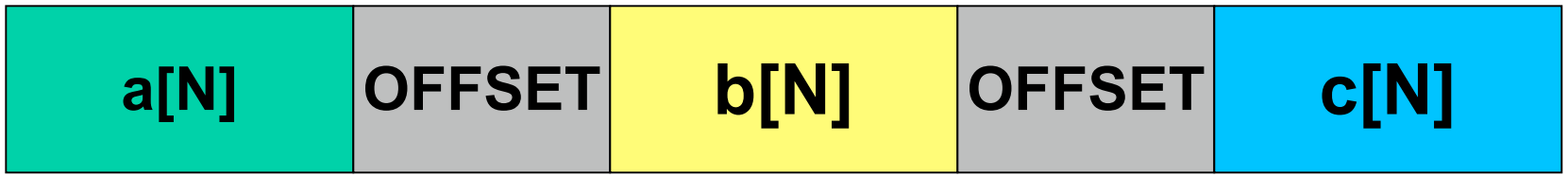
What to Expect

- Description of phenomenon we've observed using the STREAM micro-benchmark
  - Large memory bandwidth swings based on memory layout
  - Comparisons to Cray Linux Environment (CLE / CNL)
- Due to level of locality you probably aren't aware of
  - Hopefully interesting
  - Possibly useful
- Mitigation techniques we're working on that alleviate issue while maintaining LWK advantages
  - Predictable memory layout
  - Simple network stack (no pinning/unpinning)

Sandia National Laboratories

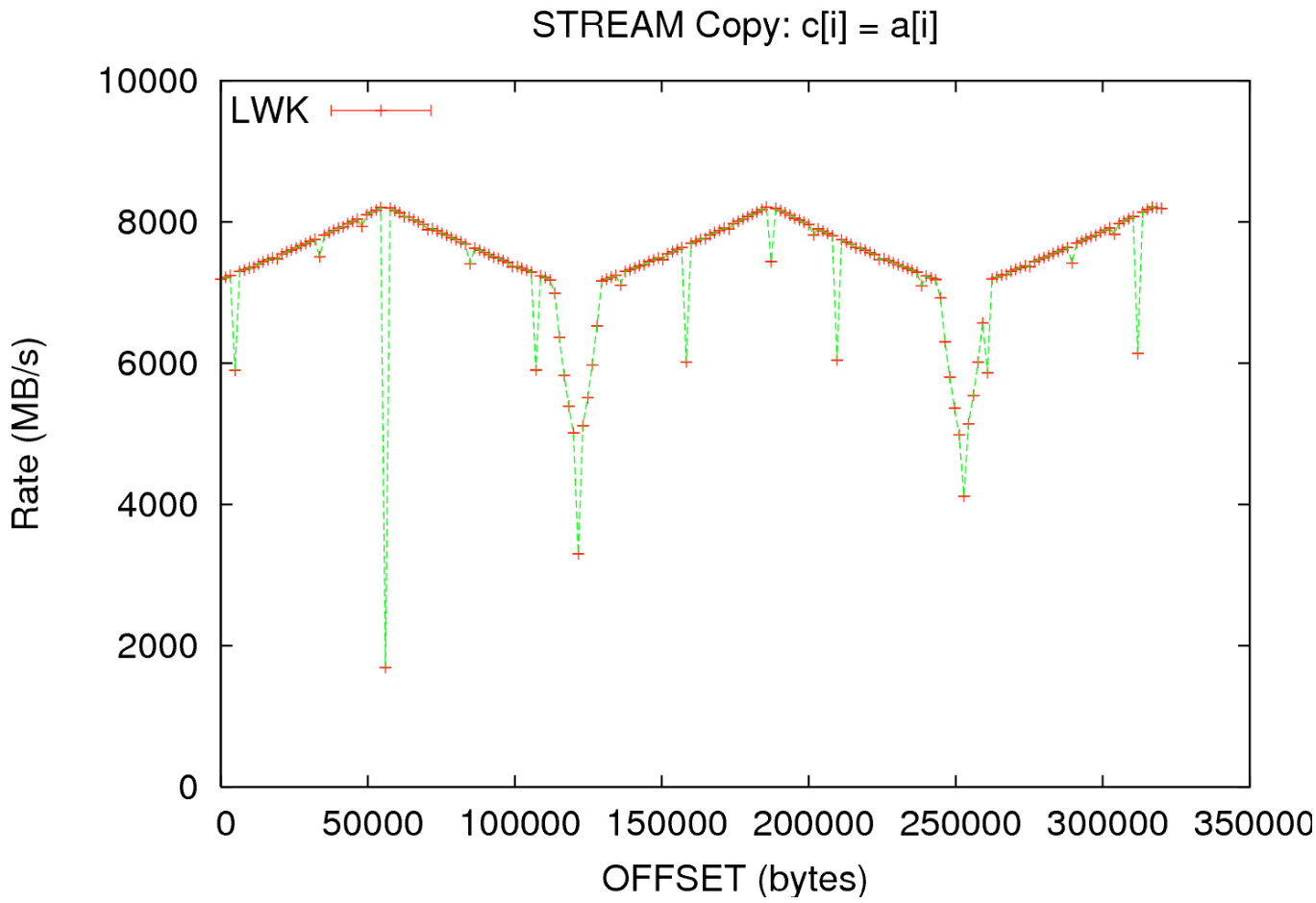
STREAM Benchmark

- Old benchmark, now component of HPCC
- Four memory intensive kernels over arrays of doubles:
  - Copy: a[i] = b[i]
  - Scale: a[i] = scalar * b[i]
  - Add: a[i] = b[i] + c[i]
  - Triad: a[i] = b[i] + scalar * c[i]
- OFFSET define controls spacing/alignment of arrays in memory:



Sandia National Laboratories

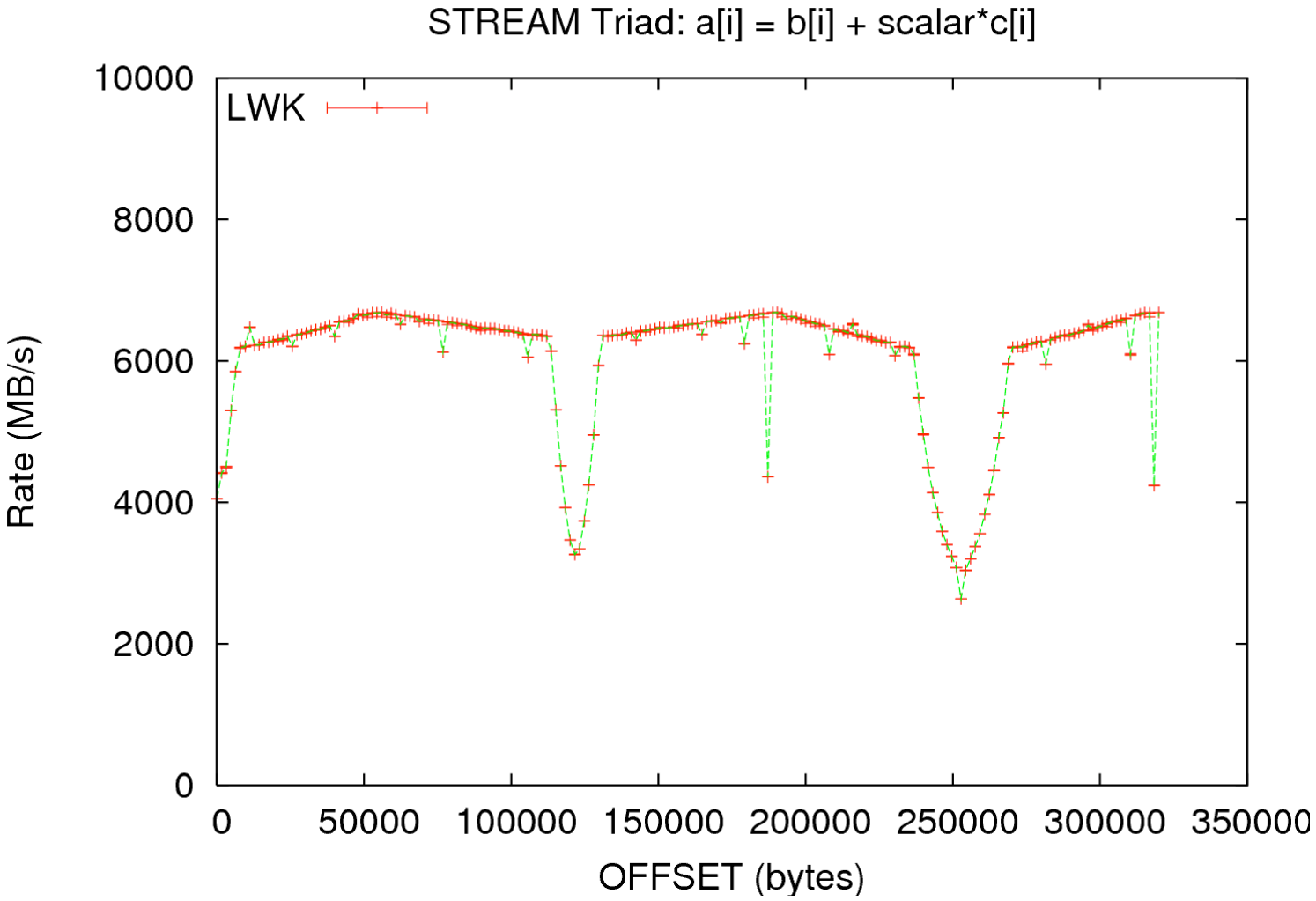
Mysterious
STREAM Copy Sawtooth on Catamount



N=2000000, ~16MB arrays

Sandia National Laboratories

STREAM Scale, Add, and Triad Similar

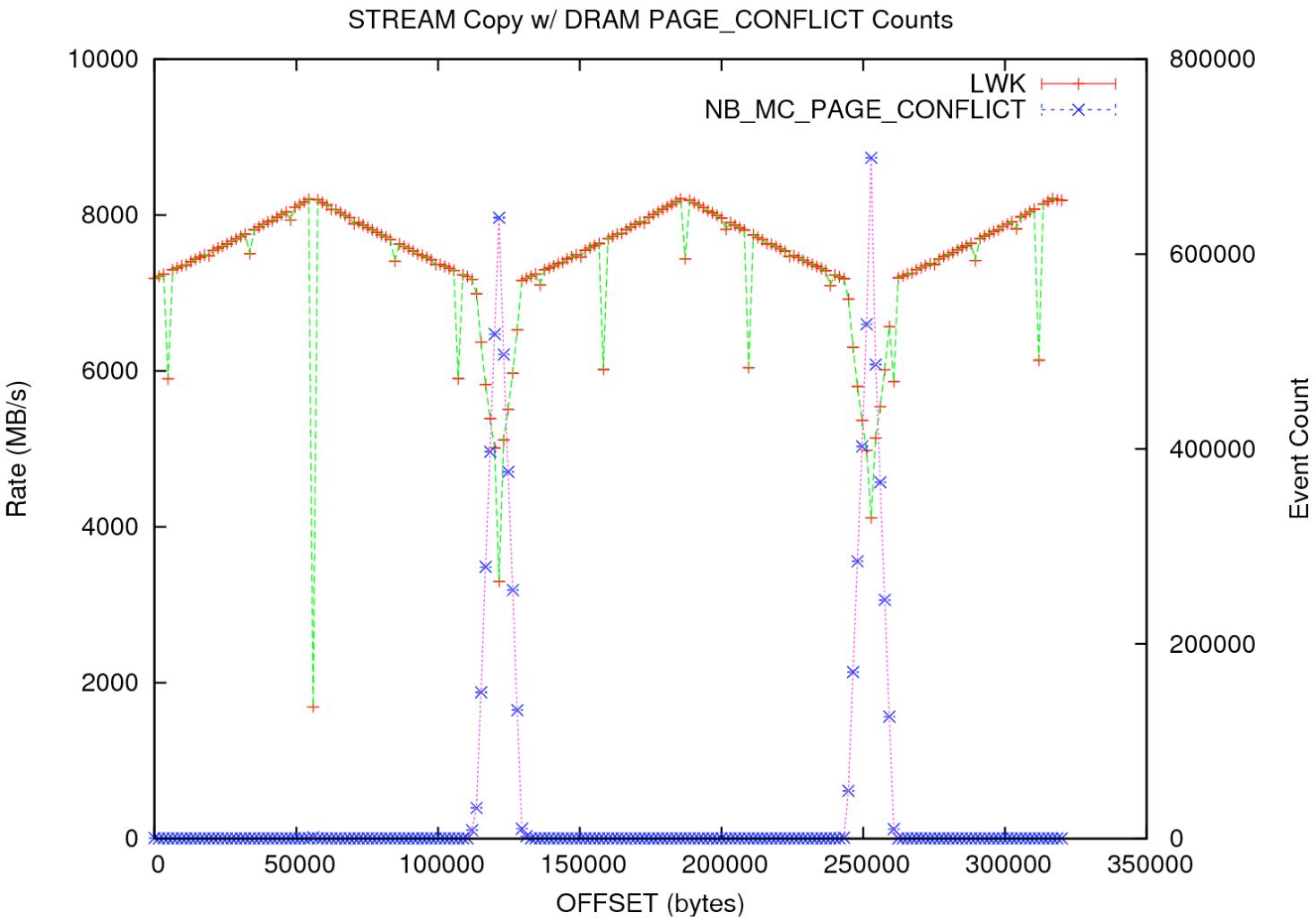


Sandia National Laboratories

What's Going On?

- Mystery for 2+ years
  - First observed by Courtenay Vaughan while gathering Red Storm HPCC results
  - Careful tuning performed to avoid valleys
- Suspects:
  - Cache aliasing?
  - Prefetch issues?
  - Non-temporal prefetch/store issues?
  - Coldstart configuration of memory controller?
  - Something inherit in Catamount?

Dips Due to DRAM Page Conflicts (Bank Conflicts)



Sandia National Laboratories

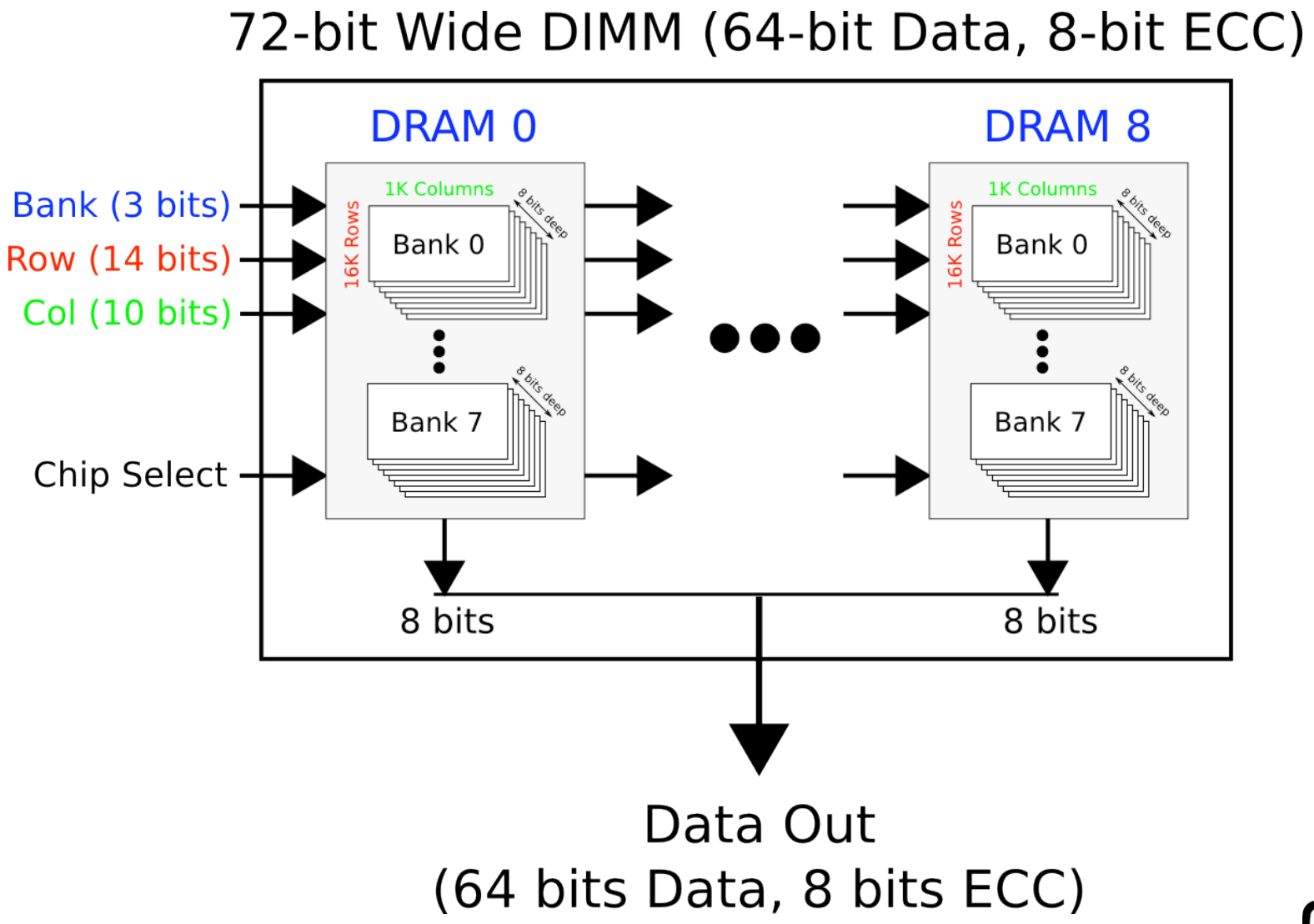
A (Very) Brief DRAM Overview

- Commodity component, most numerous in system
- 2-D array of memory
  - Addressed by (row, column, bank)
  - Accesses to different rows of same bank conflict
  - Conflicts are slow, prevents request pipelining
- Typical row (aka page) sizes:
  - DRAM: 1 KB wide (1K columns, each 8-bits deep)
  - DIMM: 8 KB wide (8 DRAM chips in parallel)

- See “Memory Systems: Cache, DRAM, Disk” book

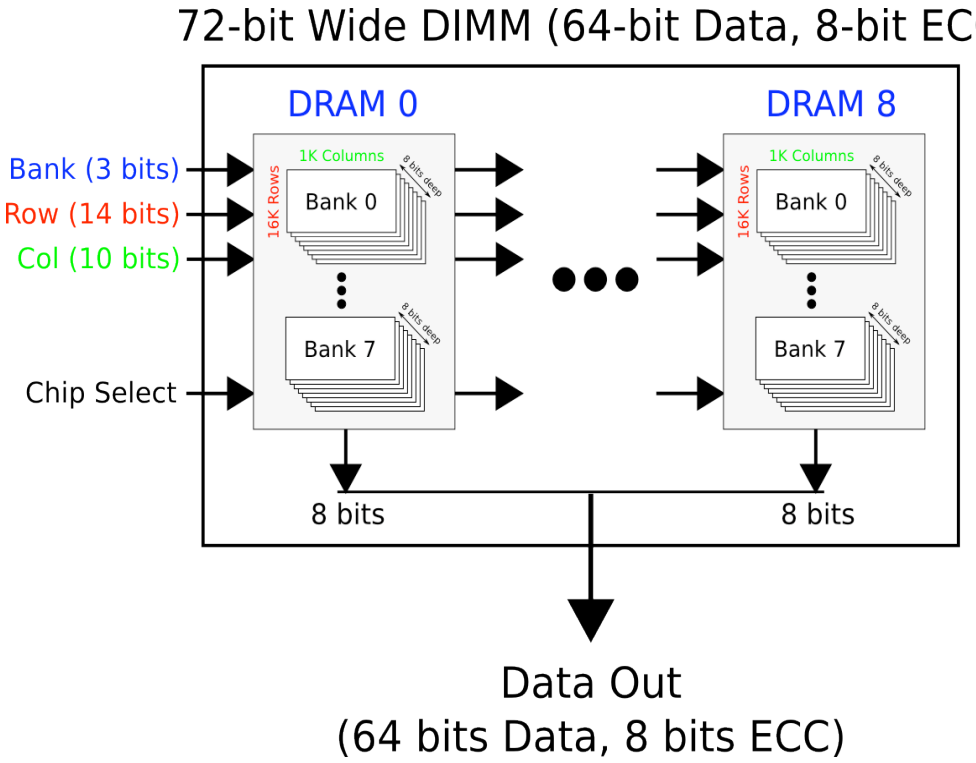
Sandia National Laboratories

DDR2 DIMM Architecture Example



Sandia National Laboratories

Red Storm DDR2 DIMM Architecture



Each DRAM Row is
1K columns * 8 bits = 1K bytes

Each DIMM Row is
1K bytes * 8 chips = 8K bytes

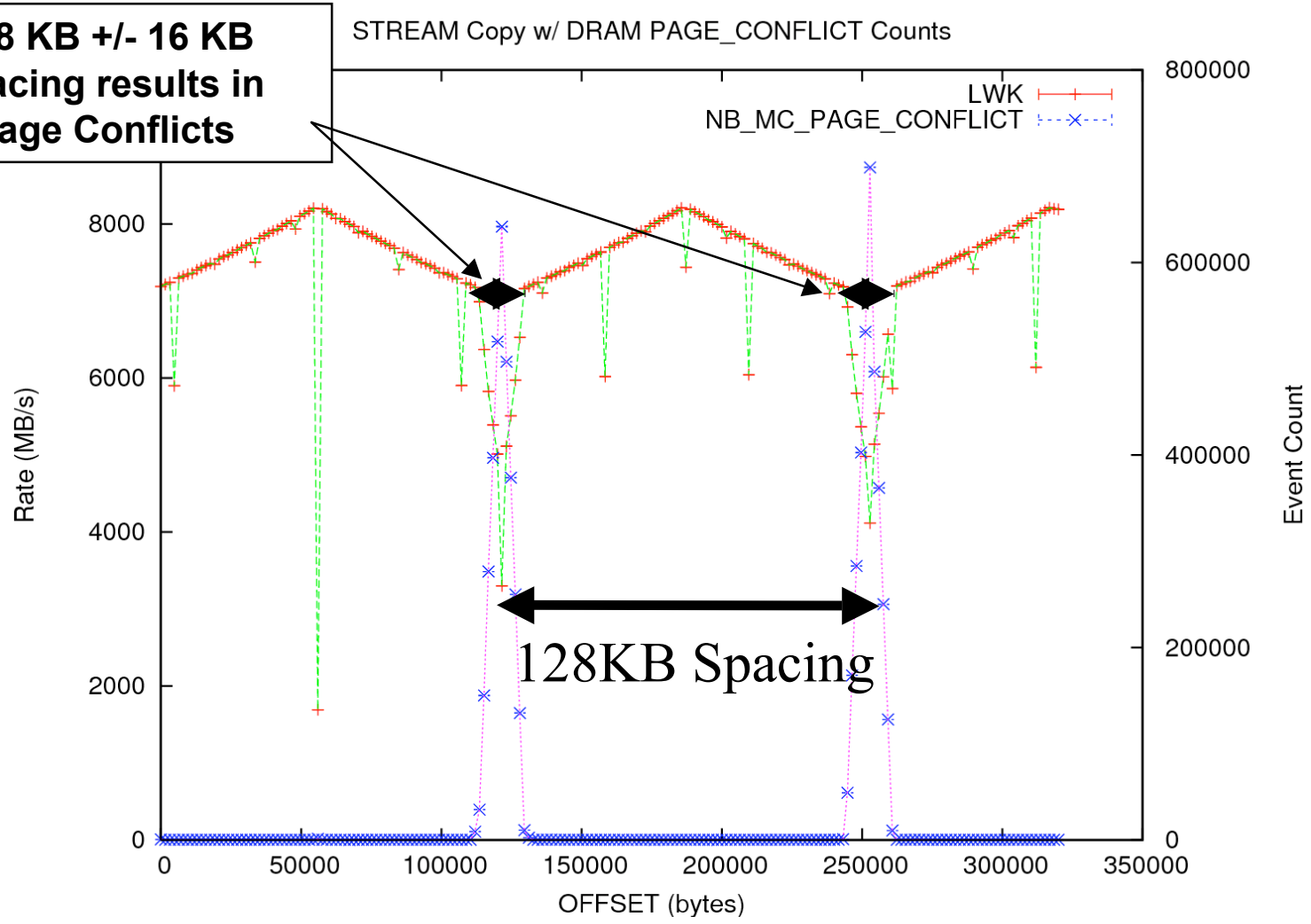
Each Memory "Page" is
8K bytes * 2 DIMMs = 16K bytes

Addresses that are
16K bytes * 8 banks = 128K bytes
apart will result in a Bank Conflict
(Consecutive accesses to
different rows in same
bank, aka Page Conflict)

Sandia National Laboratories

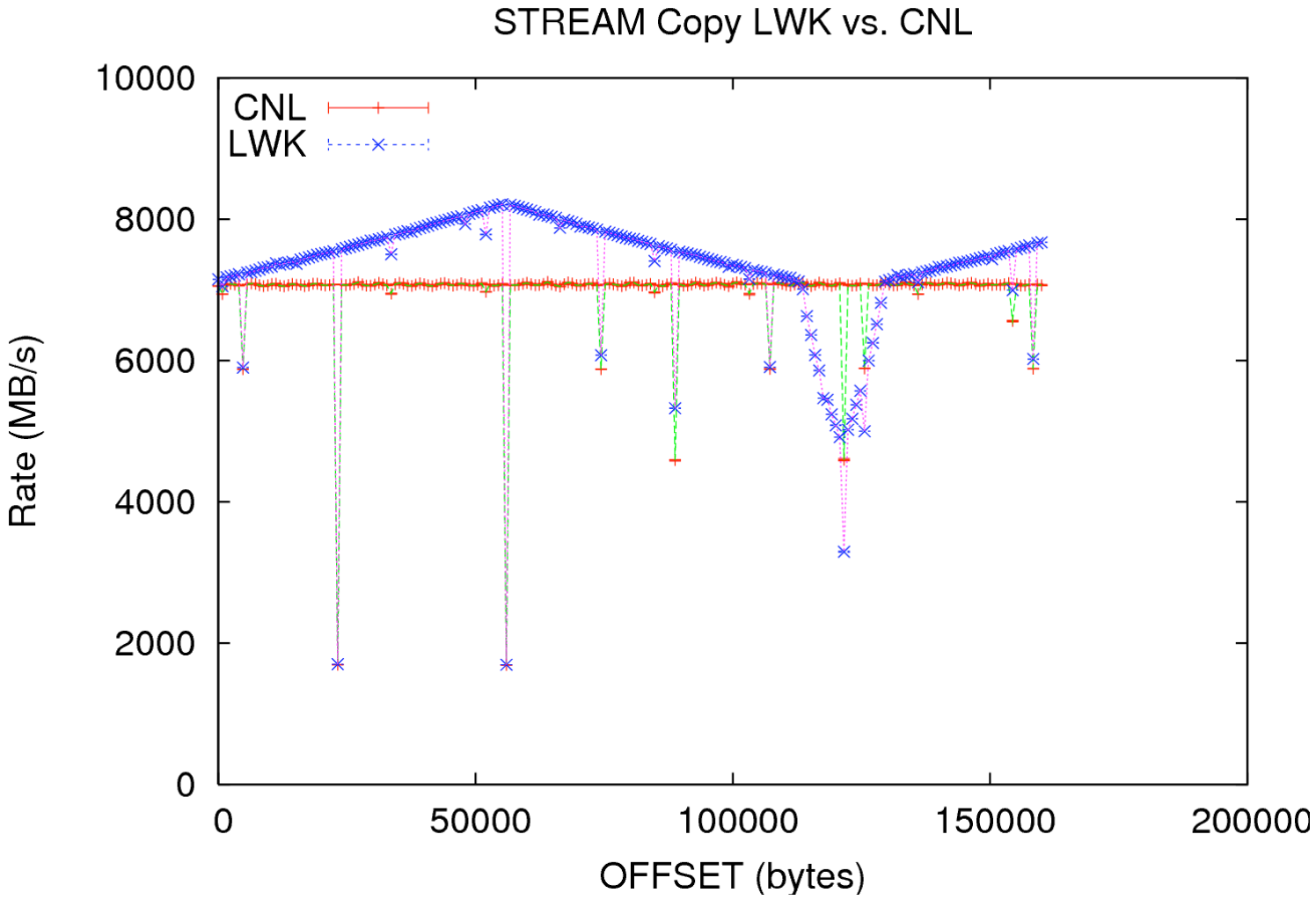
By the Numbers ...

128 KB +/- 16 KB spacing results in Page Conflicts



Sandia National Laboratories

What About Compute Node Linux?



Sandia National Laboratories

Linux Translation Strategy

- Will scatter virtual pages throughout the physical space
- Mapping is non-deterministic and varies from run-to-run

Sandia National Laboratories

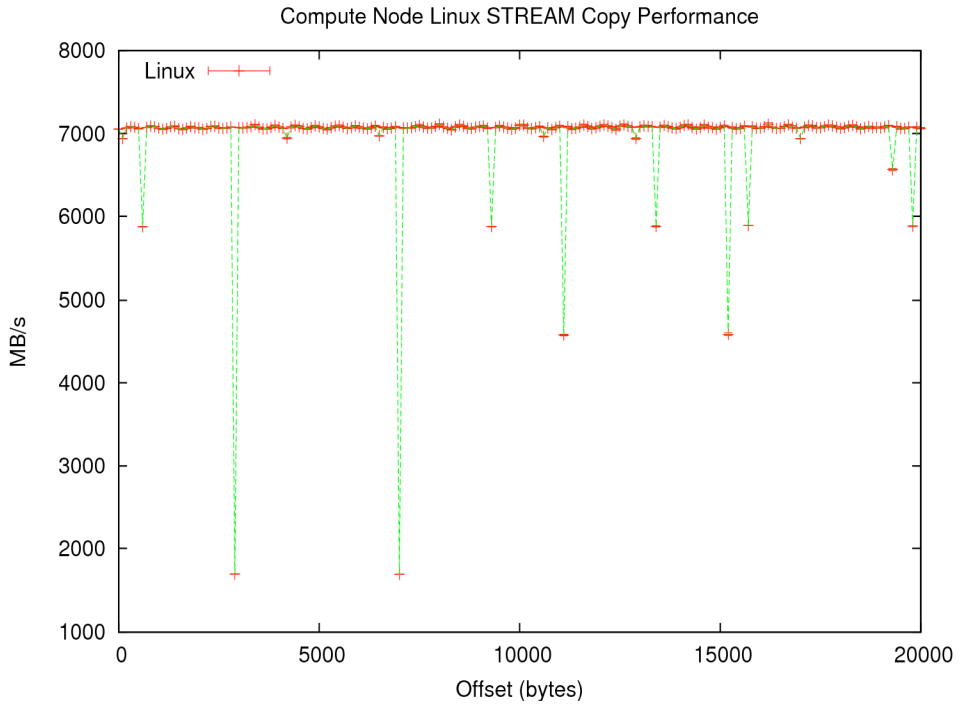
Catamount Translation Strategy

- Maps the virtual address range to a contiguous physical address range
- Done to reduce state required for SeaStar NIC

Sandia National Laboratories

Compute Node Linux Numbers

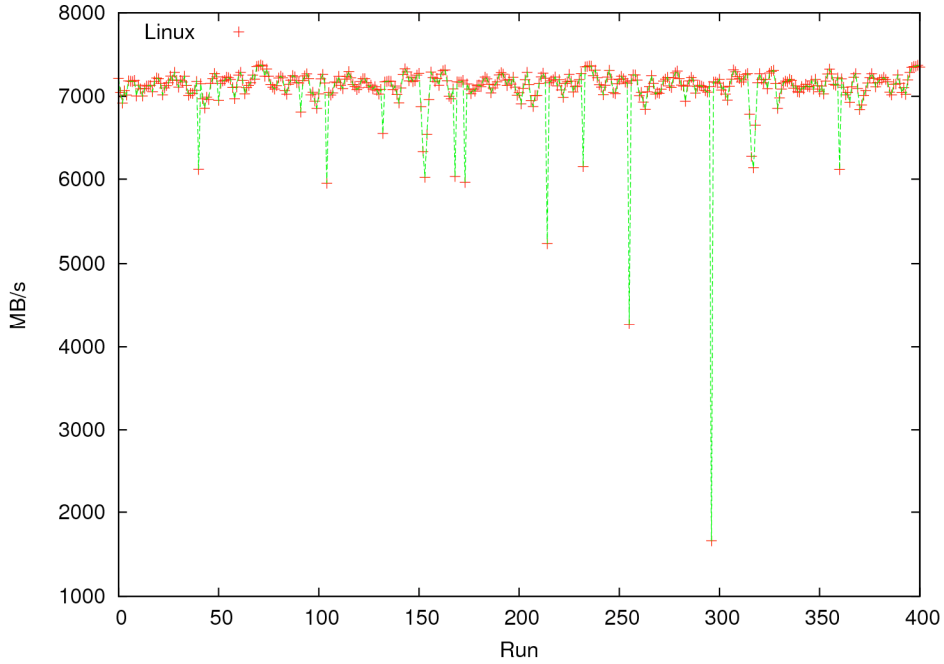
- Each point from a freshly booted CNL node
- Dips from cache aliasing and also seen on Catamount



Sandia National Laboratories

As Memory Fragments, Performance Affected

- Translations vary for each application run
- Worst case 80% slowdown due to buffer conflicts and cache aliasing
- Average case similar to best case



Sandia National Laboratories

Research Questions

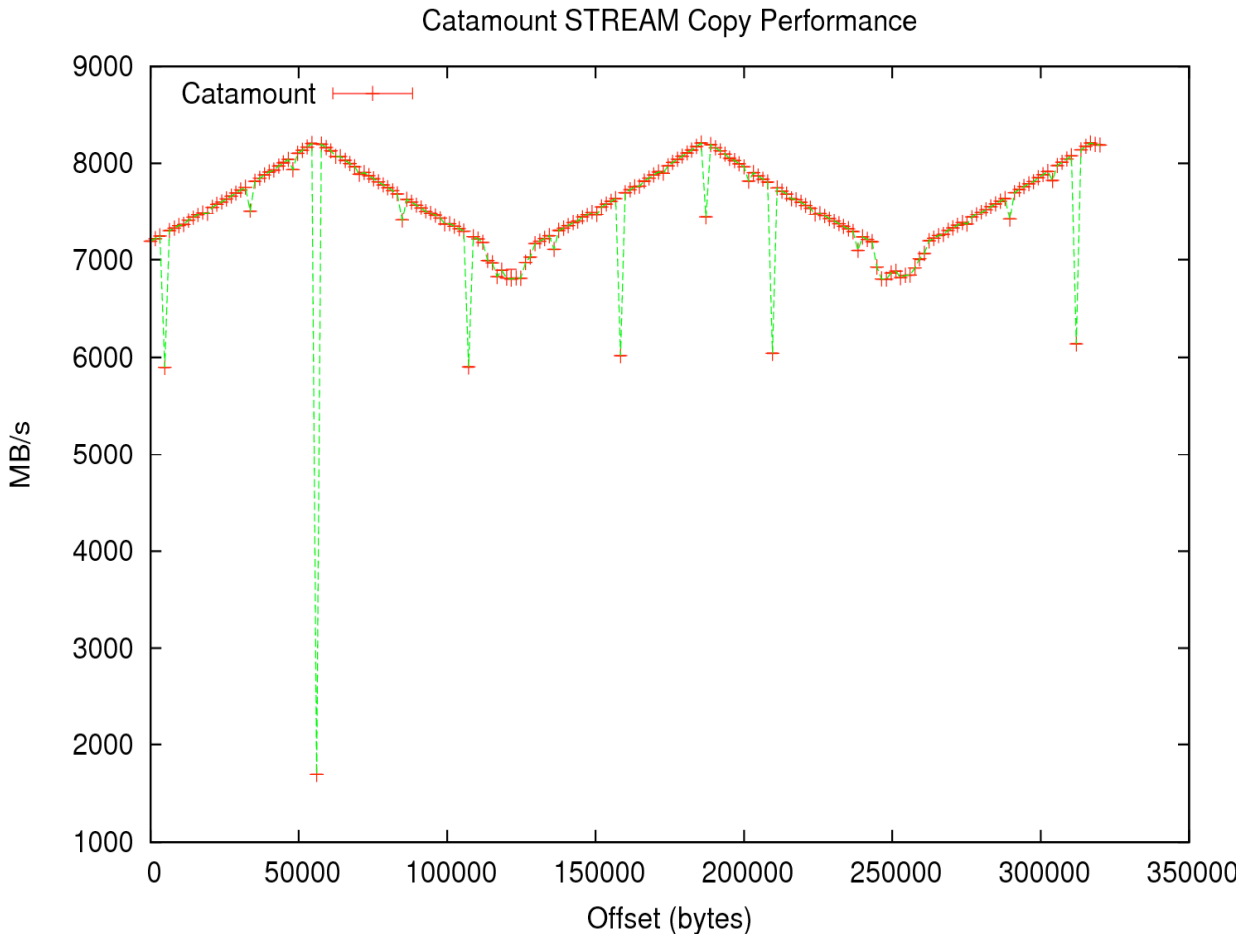
- Do page conflicts matter for any real applications?
  - Potential cause of the observed CNL vs. Catamount performance differences on Red Storm?
- Mitigation techniques:
  - Opteron memory controller “swizzle” mode
  - Randomize virtual->physical mapping
  - Deterministic virtual->physical mapping
    - No page pinning/unpinning
    - Send address/length to SeaStar vs. command array
  - Compiler optimization?
  - Stream-style programming…
    1 array with unit stride cannot cause bank conflict

Sandia National Laboratories

Adaptive Approaches

- Monitor page conflict counts while an application runs
- If system sees application page conflict counts increasing, shuffle memory mapping
- Intension: cap the number of page conflicts at a certain level

Adaptive Page Mapping Performance



Sandia National Laboratories

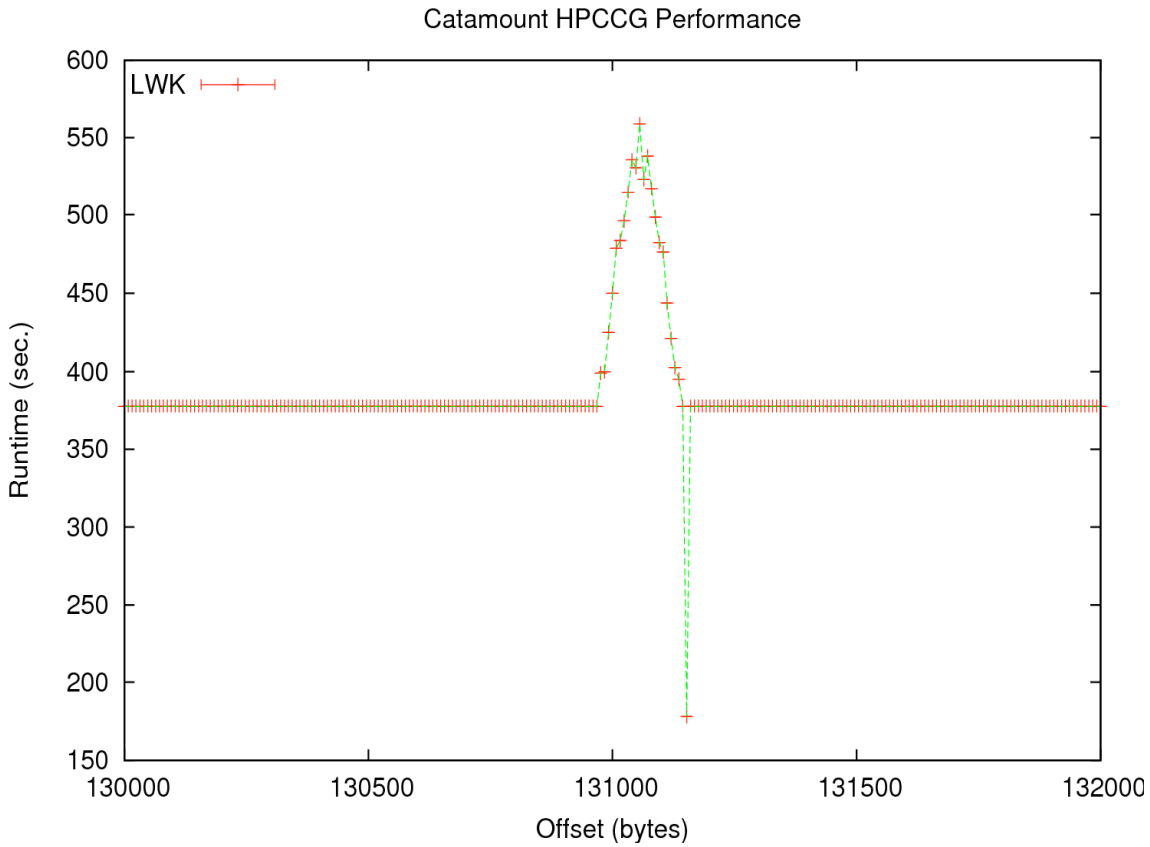
What About Real Applications?

- HPCCG: somewhere between a micro-benchmark and a real application
- Written by Mike Heroux of Sandia National Labs
- Simple preconditioned conjugate gradient solver
- Generates a 27-point finite difference matrix with a user-prescribed sub-block size on each processor
- Processor domains are stacked in the z-dimension

Sandia National Laboratories

HPCCG – Page Conflict Slowdown

- 32 nodes
- Offset identical on each node
- ~50% slowdown



Sandia National Laboratories

Summary

- Virtual to physical translations can affect the performance of HPC applications
- DRAM page buffer is another level of locality in the memory hierarchy that the programmer has little control over and may be important to application performance
- No translation strategy clear winner

Experimental Platform

- Hardware
  - 32 node Cray XT3/4 dev system at SNL
  - 2.4 GHz, dual-core AMD Opteron w/ 4 GB RAM
  - Cray SeaStar NIC
- Software
  - Catamount lightweight OS
  - Cray Compute Node Linux

Sandia National Laboratories