CSCS

Swiss National Supercomputing Centre

TLB Entries and Application Performance

Neil Stringfellow

CSCS - Swiss National Supercomputing Centre

Overview

- CSCS and the Paul Scherrer Institute bought an 1100 processor Cray XT3 system in early 2005
- A set of performance benchmarks were in the procurement and Cray committed numbers to these benchmarks
- On machine delivery, it was clear that the benchmark results could not be achieved with the default setup
- A small page size option was introduced which allowed most of the benchmark figures to be achieved
- The small page option has benefited many other applications
- Some algorithms do have small benefits from large pages

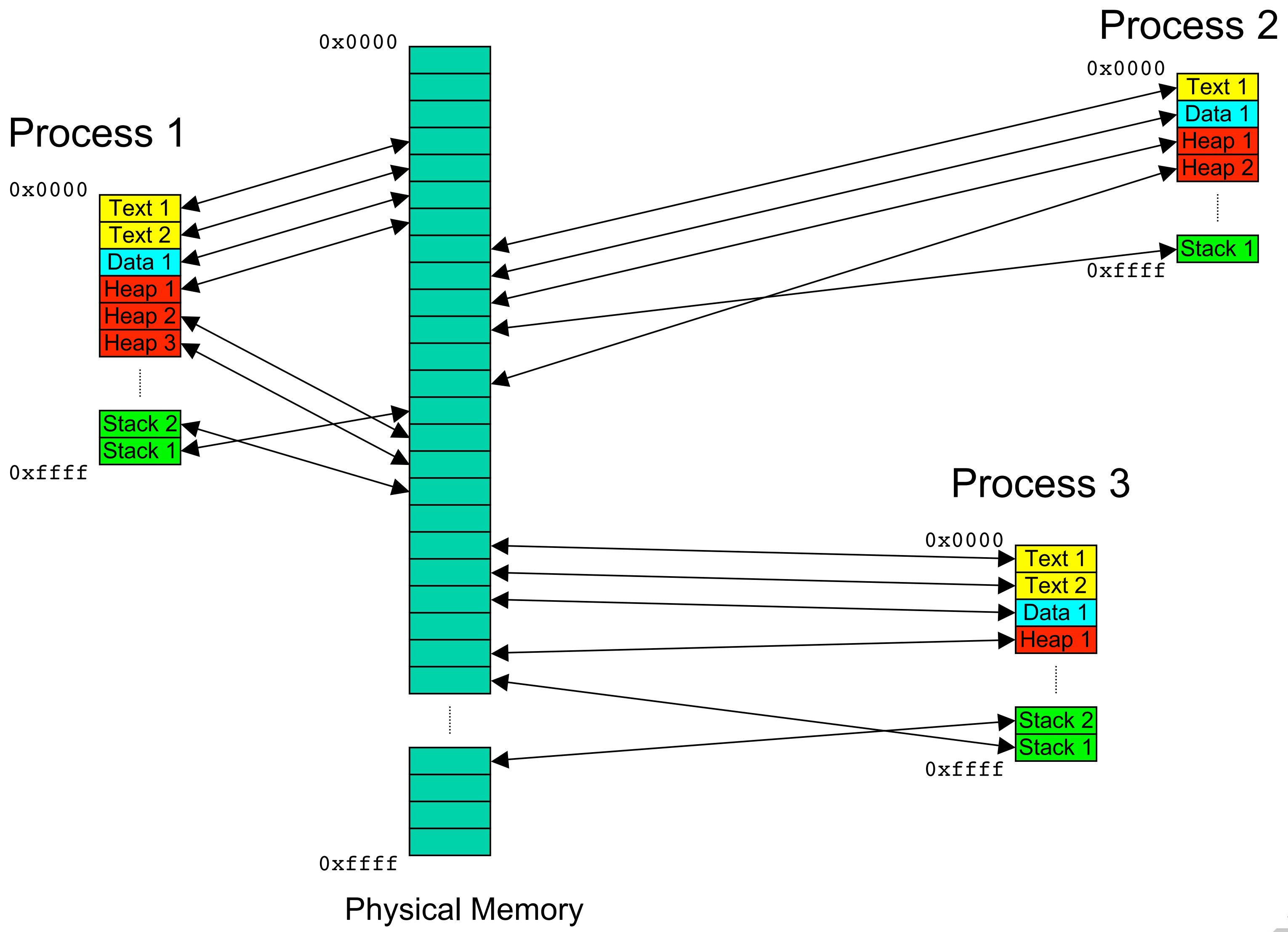
Introduction to Virtual Memory

- A process requires sections of memory in order to be able to execute
  - Text section - executable
  - Data section
  - A memory heap
  - A stack
- These sections are given addresses in memory at compile time and run time, for example
  - Text - 0x0000-0x1fff
  - Data - 0x2000-0x2fff
  - Heap - 0x3000-0xefff
  - Stack - 0xf000-0xffff
- All processes will have the same (or similar) addresses
  - This would lead to overlapping memory between processes

Virtual to Physical Mapping

- The process requests memory from the operating system
- The operating system allocates real memory to the process
- The operating system gives the process an address which the process should use to access the memory
  - This is the virtual address
- The operating system also gives the CPU a virtual to physical mapping so that the CPU can access the real physical memory
  - This is the virtual to physical mapping
- Memory is given out in “pages” of a specific size

Example of Virtual to Physical Map



Translation Lookaside Buffer

- The TLB is a part of the processor which holds translations from virtual to physical memory
- If the processor tries to access a page of memory which has its address in the TLB then the translation can occur and the memory operation can proceed
- A TLB miss occurs when the processor tries to access a memory page for which no translation is in the TLB
    - A page fault occurs and one entry in the TLB needs to be replaced with the required virtual to physical mapping
- TLB misses are very costly !!!

Page Size and TLB on the Opteron

- AMD Opteron has 2 page sizes
  - Small page of 4 Kilobytes
  - Large page of 2 Megabytes
- Processors have limited numbers of TLB entries
  - TLB entries take up “real estate” on the processor
- For the AMD Opteron there are different numbers of TLB entries for different page sizes
  - Small pages have 64 TLB entries in primary cache and 1024 entries in the secondary cache
  - Large pages have 8 TLB entries in primary cache and *NO* entries in secondary cache

Red Storm Design Choice

- For Linux the default page size is 4 kilobytes
- For Catamount, Sandia decided to use 2 Megabyte pages for the default
- Increases the memory footprint contained in TLB from 4 Megabytes to 16 Megabytes
- Minimises TLB misses if an application accesses “close” memory locations
  - Stream benchmark accesses 3 or 4 arrays
- Increases TLB misses if an application accesses many well separated memory locations
  - Accessing more than 8 arrays which are each separated by 2 Megabytes

CSCS Procurement Benchmarks

- CSCS Benchmarks were
  - IOZone (8 tests)
  - HPC Challenge (23 tests)
  - MoldyPSI - Molecular Dynamics Code (2 tests)
  - Trilinos based applications (8 tests)
- Cray used a cluster to get basic benchmarks and extrapolated to committed figures for the XT3
- Cray were allowed to miss up to 8 of their committed numbers with consequent financial penalties
- With large pages Cray were due to miss 12 of their targets

Code Analysis

- Roberto Ansaloni (Cray benchmarker) discovered high numbers of TLB misses on user benchmark codes
- Claudio Redaelli saw similar numbers of TLB misses on key CSCS computational chemistry codes
  - Mainly in the area of classical molecular dynamics
- Other large numbers of TLB misses were seen on codes at CSCS
- High priority was given to getting the small page/high TLB entry facility of the Opteron to be usable from Catamount
  - The Acceptance could not be passed without it
  - CSCS' users would have poor performance without it

A New Option to `yod`

- `yod` is the process which services a computational job
  - launches a job on the compute nodes
  - controls the job
  - services system requests from the job
- A new option was provided to `yod` to allow the small page size to be selected
  - option is `-small_pages`
  - real purpose of option is `-lots_of_TLB_entries`
  - the real changes to allow this option were made in the catamount kernel
- This made dramatic improvements in many codes

Acceptance Codes

- IOZone was able to be passed with either page size
- The small page option meant that 6 of the 10 user benchmarks could be passed
  - 4 Trilinos benchmarks for sparse solver still failed
- One extra HPC Challenge benchmark could also be passed with small pages
  - This was unexpected and therefore saved Cray some money

Effect on HPC Challenge

- Summary of results from HPC Challenge for change from large to small page size
  - Single Node/Embarrassingly Parallel were worse
    - Stream Benchmark - 1-2% worse
    - Random access on node - over 40% worse
    - DGEMM - 1% worse
    - FFT - 10% worse
  - Global Benchmarks were generally better
    - HPL - 1% worse
    - Ptrans - 3% better
    - Random access - 5% better
    - Latency - 4-5% better
    - Bandwidth - about the same
    - FFT - 2% better
- Random Ring Latency improvement allowed performance test to be passed

Other Codes

- LM Numerical weather prediction code
  - The LM_RAPS public version showed an improvement of over 10% on 28 processors
    - This was true for low resolution and high resolution cases
  - This 10% improvement was also seen with the full aLMo code used by MeteoSwiss on over 400 processors
- Some slight degradation for dense linear algebra dominant codes
  - Approximately 1-6% degradation on ScaLAPACK routines

Molecular Dynamics Codes

- Standard test suite for Orac showed a reduction of 10-20% in execution time
- Tests from Amber and NAMD showed even higher reductions up to 40% improvement
- Most of CSCS' molecular dynamics codes are now run with the `-small_pages` option
  - Still about 30% of runs are done without the option
- Classical molecular dynamics packages account for about 15-20% of runtime on the Cray XT3 at CSCS
  - Accounting from NAMD, Amber, Orac, DL_POLY, MoldyPSI

Another Example

- Mike Ashworth from Daresbury Laboratory was running a turbulence code on the XT3 at CSCS

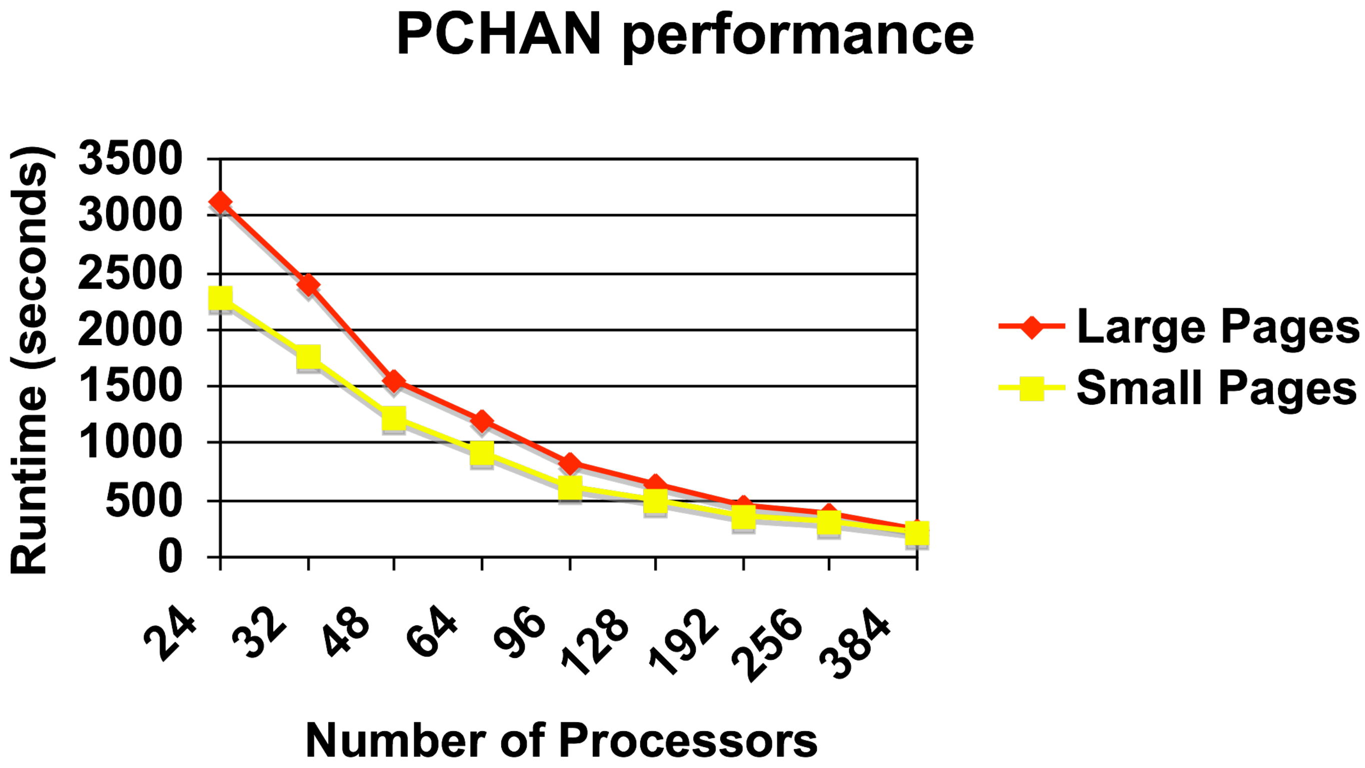
  Me: How did your runs go on the XT3 ?

  Mike: Here’s a graph (XT3 worst of 3 machines)

  Me: Could you rerun with the `-small_pages` option as I'm wondering whether this PCHAN could be the first code to perform worse with small pages

  Mike: Sorry, small pages is good for PCHAN as well.

PCHAN Improvement



Conclusions

- The -small_page option has been an improvement for many codes on CSCS' Cray XT3
- Usage of the option has increased and scientific throughput for many groups has improved
- Many people who use their own codes do not make use of `-small_pages` and probably do not try the option either
- It is important to remember that this is not an issue of page size but of numbers of TLB entries

Recommendations

- The default for small or large pages when `yod` launches a job should be a site-configurable default
  - In something like `/etc/yod.conf`
  - Needs also a `-large_pages` option to `yod`
- Work on Cray supplied numerical libraries should also be done with both small and large pages
  - There might not be many libraries which this should affect
  - If Cray ships the Goto BLAS, page size could be significant
- AMD should put a large number of TLB entries for large pages on the Opteron

Acknowledgements

- Thanks to Roberto Ansaloni, Maria Grazia Giuffreda, Claudio Redaelli, Mauro Ballabio, Kevin Roy, Craig Lucas and Mike Ashworth for providing benchmark results
- Thanks to Roberto Ansaloni for the initial work in identifying the problem of high numbers of TLB misses with the default catamount page size
- Thanks to Roberto Ansaloni and Mario Mattia for pursuing the need for the `-small_pages` option within Cray