Performance Study of Hyper-Threading Technology on the LUSITANIA Supercomputer

César Gómez-Martín, José Luis González-Sánchez, Javier Corral-García, Angel Bejarano-Borrega, Javier Lázaro-Jareño

CénitS (Centro Extremeño de iNvestigación, Innovación Tecnológica y Supercomputación)
Fundación COMPUTAEX

4th Iberian GRID Infrastructure Conference, IBERGRID 2010 - BRAGA

Outline

1. Introduction
   - Hyper-Threading Technology
   - LUSITANIA
   - Motivations of the study
2. Performance Evaluation
   - High Performance Linpack (HPL 1.0a)
   - NAS Parallel Benchmark - Fast Fourier Transform
   - NAS Parallel Benchmark - Embarrassingly Parallel
   - NAS Parallel Benchmark - Block Tridiagonal
   - NAS Parallel Benchmark - Conjugate Gradient
3. Summary
   - Conclusions
   - Recommendations
   - Thanks!

Outline

1. Introduction
   - Hyper-Threading Technology
   - LUSITANIA
   - Motivations of the study
2. Performance Evaluation
   - High Performance Linpack (HPL 1.0a)
   - NAS Parallel Benchmark - Fast Fourier Transform
   - NAS Parallel Benchmark - Embarrassingly Parallel
   - NAS Parallel Benchmark - Block Tridiagonal
   - NAS Parallel Benchmark - Conjugate Gradient
3. Summary
   - Conclusions
   - Recommendations
   - Thanks!

Hyper-Threading Technology

Hyper-Threading technology is based on Simultaneous Multi-Threading (SMT), which is a combination of:

- Instruction Level Parallelism (ILP): increase the number of instruction that can be executed during one clock cycle.
- Task Level Parallelism (TLP): allows multiprocessor systems to concurrently execute several tasks from one or various programs.

Outline

1. Introduction
   - Hyper-Threading Technology
   - LUSITANIA
   - Motivations of the study
2. Performance Evaluation
   - High Performance Linpack (HPL 1.0a)
   - NAS Parallel Benchmark - Fast Fourier Transform
   - NAS Parallel Benchmark - Embarrassingly Parallel
   - NAS Parallel Benchmark - Block Tridiagonal
   - NAS Parallel Benchmark - Conjugate Gradient
3. Summary
   - Conclusions
   - Recommendations
   - Thanks!

LUSITANIA

- LUSITANIA is a SMP-ccNUMA system with 2 HP SuperDomes SX2000 nodes installed at the Supercomputing Center of Extremadura (CénitS)

Hardware Configuration

Two shared-memory HP Integrity SuperDome SX2000 supernodes with:

- 64 dual-core Intel Itanium2 Montvale processors at 1.6GHz with 18 MB cache and 1TB memory on a single image.
- Architecture is based on explicit ILP. Compiler makes the decisions about which instructions will execute in parallel.
- Up to six instructions per clock cycle.

Hardware Configuration

Two shared-memory HP Integrity SuperDome SX2000 supernodes with:

- SX2000 chipsets are interconnected via crossbar switches with three independent connections to ensure the best performance by using multipathing, ECC protection and load-balancing.
- 16 cells with interleaved memory for shared objects or data structures with uniform latency.
- The cell controller (CC) maintains a cache-coherent memory system (ccNUMA) using a directory-based memory controller

Software Configuration

- Suse Linux Enterprise (v.10).
- Intel Fortran Compiler (v.11.0.074), O3 level of optimization.
- Intel C/C++ Compiler (v.11.0.074), O3 level of optimization.
- HP-MPI libraries and binaries (v.02.02.07.00 Linux IA64).
- Intel OpenMP implementation to generate multi-threaded applications.

Outline

1. Introduction
   - Hyper-Threading Technology
   - LUSITANIA
   - Motivations of the study
2. Performance Evaluation
   - High Performance Linpack (HPL 1.0a)
   - NAS Parallel Benchmark - Fast Fourier Transform
   - NAS Parallel Benchmark - Embarrassingly Parallel
   - NAS Parallel Benchmark - Block Tridiagonal
   - NAS Parallel Benchmark - Conjugate Gradient
3. Summary
   - Conclusions
   - Recommendations
   - Thanks!

Motivations of the study

Hyper-Threading was disabled OOTB:

- Demanded applications are multidisciplinary and heterogeneous.
- One of the first paradox was to investigate whether the effects of Hyper-Threading were good or harmful.
- It is very important to clarify which is the best configuration to slightly adjust all the parameters that could improve system performance.
- Some bechmarks have been chosen to try to demonstrate the performance gain or degradation when running dfferent kind of applications.

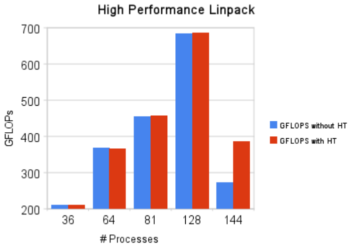
Outline

High Performance Linpack

- It is a performance test widely used by High-Performance Computing community
- It evaluates a very specific performance area by solving a random dense system of linear equations in double precision (64 bits) arithmetic
- Provides a quite accurate way of knowing the performance of real applications on a HPC environment

High Performance Linpack

# Processes	GFLOPS without HT	GFLOPS with HT
36	211.0	210.5
64	368.4	366.3
81	456.0	457.7
128	685.2	686.3
144	274.3	386.3



High Performance Linpack

- There aren't big improvements if we enable Hyper-Threading.
- When the number of processes is less than the number of logical processors the performance is similar.
- Only when the system is overloaded we achieve around 40% performance improvement.
- HPL was compiled using the BLAS library from the Intel Math Kernel Library, the floating point units were very optimized and didn't allow much space for concurrent processes on the same unit.

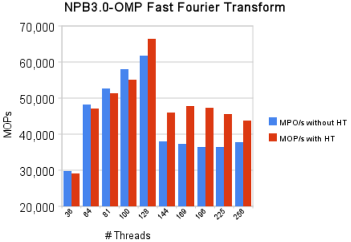
Outline

NAS Parallel Benchmark - Fast Fourier Transform

- This benchmark contains a kernel of a 3-D fast Fourier Transform (FFT)-based spectral method.
- FT performs three one-dimensional FFTs, one for each dimension.
- It is a rigorous test of long-distance communication performance.

NAS Parallel Benchmark - Fast Fourier Transform

# Threads	MOPS without HT	MOPS with HT
36	29780.29	29195.41
64	48249.32	47119.27
81	52654.43	51265.21
100	58083.44	55094.50
128	61852.50	66353.47
144	38097.25	46065.66
169	37318.60	47670.45
196	36430.32	47253.03
225	36468.69	45490.30
256	37794.78	43818.99



NAS Parallel Benchmark - Fast Fourier Transform

- When we use all the processors, HT improve the performance more than 7%.
- Whenever the system is overloaded the improvement of HT is between 15% and 29%.
- The design of the FT benchmark requires using intensive floating point operations, but it also needs communication among processes.

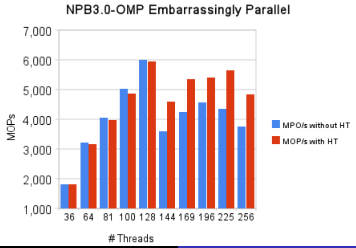
Outline

NAS Parallel Benchmark - Embarrassingly Parallel

- Generates pairs of Gaussian random deviates according to a specific scheme
- The goal is to establish the reference point for peak performance of a given platform
- It provides an estimate of the upper achievable limits for floating point performance without significant interprocessor communication

NAS Parallel Benchmark - Embarrassingly Parallel

# Threads	MOPS without HT	MOPS with HT
36	1816.52	1820.17
64	3210.63	3175.16
81	4048.68	3980.22
100	5023.44	4868.53
128	6008.92	5959.08
144	3589.57	4590.48
169	4229.76	5364.78
196	4563.59	5407.18
225	4347.38	5646.47
256	3763.56	4839.71



NAS Parallel Benchmark - Embarrassingly Parallel

- This problem is used in all the typical Monte Carlo simulations, threads don't need to communicate very often, they just calculate arithmetic operations.
- HT disabled provides 1%-3% MOPS gain.

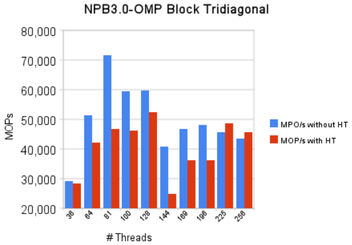
Outline

NAS Parallel Benchmark - Block Tridiagonal

- It is a simulated CFD application that uses an implicit algorithm to solve 3-dimensional (3-D) compressible Navier-Stokes equations
- The finite differences solution to the problem is based on an Alternating Direction Implicit (ADI) approximate factorization that decouples the x, y and z dimensions
- The resulting systems are Block-Tridiagonal of 5x5 blocks and are solved sequentially along each dimension

NAS Parallel Benchmark - Block Tridiagonal

# Threads	MOPS without HT	MOPS with HT
36	29194.51	28448.64
64	51415.60	42080.26
81	71627.49	46754.70
100	59387.02	46164.66
128	59781.11	52312.25
144	40925.23	24923.54
169	46653.48	36100.99
196	48003.63	36292.36
225	45660.17	48648.60
256	43382.45	45569.99



NAS Parallel Benchmark - Block Tridiagonal

- Best performance is obtained by using 81 threads instead of 128.
- When the threads are able to exploit all the cache locality the number of MOPS increases.
- It is unwise to use HT for cache-friendly applications.

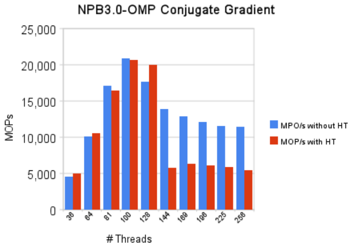
Outline

NAS Parallel Benchmark - Conjugate Gradient

- It is a method to compute an approximation to the smallest eigenvalue of a large, sparse, unstructured matrix
- It tests irregular long distance communications employing unstructured matrix vector multiplication

NAS Parallel Benchmark - Conjugate Gradient

# Threads	MOPS without HT	MOPS with HT
36	4592.96	5044.12
64	10165.17	10529.92
81	17070.85	16470.47
100	20865.44	20664.16
128	17628.43	19970.67
144	13841.09	5786.25
169	12870.31	6296.99
196	12114.52	6143.35
225	11523.90	5928.27
256	11402.77	5440.40



NAS Parallel Benchmark - Conjugate Gradient

- 100 threads is the configuration with the highest MOPS rate.
- The combination of computation and communication among threads implies that performance without HT is better in most of the cases.

Outline

Conclusions

- The sense of the application will determine whether Hyper-Threading will accelerate or diminish performance on the HP Integrity SuperDome SX2000.
- Enabling Hyper-Threading or disabling it is not an exact science, a bechmark study of the problem is highly recommended.

Outline

Recommendations

- Hyper-Threading is recommended when:
  - Threads or processes have to communicate with each other very often.
  - An MPI library is used to communicate processes.
  - The application doesn't use all the processing units leaving space for a different thread or process.
  - Poor-optimized math libraries are used, if it is not optimized by the software then it can be optimized by the hardware.
  - There are a lot of cache misses.

Recommendations

- Hyper-Threading is not recommended when:
  - The application is embarrassingly parallel, i.e. there is almost no communication between threads and processes.
  - Intensive floating point operations exploit all the processor resources and there is no more room for a different operation.
  - A cache-friendly application is executed.
  - The math library is already optimized and parallelized.

Outline

1. Introduction
   - Hyper-Threading Technology
   - LUSITANIA
   - Motivations of the study
2. Performance Evaluation
   - High Performance Linpack (HPL 1.0a)
   - NAS Parallel Benchmark - Fast Fourier Transform
   - NAS Parallel Benchmark - Embarrassingly Parallel
   - NAS Parallel Benchmark - Block Tridiagonal
   - NAS Parallel Benchmark - Conjugate Gradient
3. Summary
   - Conclusions
   - Recommendations
   - Thanks!

Thank you!

- César Gómez Martín
- cesar.gomez@cenits.es
- http://www.cenits.es