System & App Performance Work by PAL

Adolfy Hoisie

Kevin Barker, Kei Davis, Roberto Gioiosa, Song Jiang, Greg Johnson, Darren J. Kerbyson, Mike Lang, Scott Pakin, Fabrizio Petrini, Jose Carlos Sancho

Performance and Architecture Laboratory (PAL)
http://www.c3.lanl.gov/par_arch
Computer and Computational Sciences Division
Los Alamos National Laboratory
Major Thrusts

- Understanding application and system performance on present-day extreme-scale architectures through the development and application of technologies for measurement and modeling of program and system behavior,
- Devising software strategies to ameliorate application performance bottlenecks on today’s architectures,
- Modeling the behavior of applications (and systems) to understand factors affecting their scalability on future generations of extreme-scale systems, and
- Investigating software technology that will enable higher performance on next-generation, extreme-scale parallel systems.
Work since the last P&S Meeting

- **Advances in Modeling**
  - New applications modeled (Partisn, POP, HYCOM, LBMHD)
  - New systems modeled (BG/L, BG/P, XD1, Lightning, PERCS…)
  - Directly contributed to an ASCI L1 milestone
  - Single processor performance
  - Commodity memories

- **Advances in Systems**
  - Feasibility studies on incremental checkpointing
  - Early version of a kernel-level incremental checkpointer
  - Various publications on system software implemented on top of STORM primitives

- **External Activity**
  - 12 journal and top conference papers published (6 more already accepted for 2005)
  - 20+ technical reports written (more programmatic/project related)
  - Numerous presentations given at conferences/workshops
  - Invited lectures
  - Tutorial at SC’04 (50 attendees)
  - Tutorial at HPCA this week
  - Technical paper at SC’04 on BG/L – standing room only!
  - Tutorial to funding agencies, January 2005
  - In the organizing committee/program committee for many conferences (SC, Hot Interconnect, IPDPS, etc)
List of Publications (1)


List of Publications (2)


List of Publications (3)


Other technical reports/posters

- “Discussion Document 1: Modeling the PERCS networks”, LA-CP 04-094
- “LIGHTNING: PERFORMANCE RESULTS FOR THE LEVEL 2 MILEPOST”, Kei Davis, Adolfy Hoisie, Greg Johnson, Darren J. Kerbyson, Mike Lang, LA-UR 04-5064
- “A performance model of the parallel ocean program”, Darren J. Kerbyson, Phil Jones, LA-UR-04-8793
- “A note on the performance of the EM64T (Nacona) node” LA-UR-04-7450
- “PERFORMANCE MODELING AT LARGE-SCALE SYSTEMS IN PERCS: A METHODOLOGY DESCRIPTION”, LA-UR-04-5214
- “Empirical Analysis of Various Memory Models on an Intel EM64T Based Processing Node”, LA-UR-04-7449
- “A note on the difference task mappings of SAGE onto BG/L”, Darren J. Kerbyson, LA-UR-04-4118
Selected Presentations

- System and Application Performance at Extreme-Scale, invited plenary speaker at SIAM Parallel.
- Talk/briefing to the NAS panel on Supercomputing
- Invited talk at ScalPerf, Bologna, August 2004
Future Work

- Continue expanding the scope of modeling to new apps
- Include tri-Lab workload in the modeling portfolio
- Emphasize tool R&D for simplifying the modeling process
- Work on BG, Red Storm and Purple
- Single-processor modeling: interactions with Rice, Cornell, Intel, IBM, etc.
- We have a ASC Level 2 milestone in Q4 05: big pressure!
Advances in Applications Modeled
Workload Modeled

Recently Completed:
- HYCOM – Hybrid Ocean Model
- Partisn – $S_N$ transport
- POP – Parallel Ocean Program (part of CSSM)
- LBMHD – Magnetohydrodynamics

Existing:
- MCNP – Monte Carlo N-Particle
- Sweep3D – $S_N$ transport kernel on structured grids
- SAGE – hydro on AMR grids
- Tycho/UMT – $S_N$ transport on unstructured grids

In-progress:
- CICE – Sea Ice Model (part of CSSM)
- RF-CTH
LBMHD model validation

- Example Validation:
  - 32 node, 2-way Itanium-2 (Madison) 1.3GHz
  - LBMHD strong, and weak-scaling modes
  - Use of one, and both PES, per node

Max error: 11%, Average error: 2.5%
POP model validation

- Example Validation:
  - 32 node, 4-way AlphaServer ES40, 833MHz
  - 2048 node BlueGene/L, 700MHz
  - POP input decks: test, and degree 1 resolution (x1)

Max error: 4.7%  Average error: 2.0% (AlphaServer)
10.1%  4.0% (BlueGene/L)
Partisan model validation

- **Example Validation:**
  - 128 node, 4-way AlpaServer ES45 (ASCI Q type), 1.25GHz
  - Sntiming input deck
  - Model two main elements of sweep and diffusion

- **Max error:** 20%, **Average error:** 8%
Advances in Machines Analyzed
Main Machines examined

- **BlueGene/Light**
  - Performance of full-sized (64K-node) system compared with Q
  - Talk/paper presented at SC’04

- **BlueGene/P**
  - Possible future IBM product
  - Models have been used to examine expected performance
  - Report produced August ’04 (proprietary LANL/IBM)

- **IBM PERCS (DARPA HPCS)**
  - Preliminary performance analysis undertaken
  - Examining performance sensitivity of sub-systems on overall performance
  - Results being presented at DARPA review, Feb ‘05
Main Machines examined

- **Lightning**
  - Performance measured and modeled during ’04

- **Cray XD1**
  - Preliminary analysis of XD1 system undertaken
  - Communication performance examined
  - Performance of SAGE and Sweep3D modeled
  - Work presented at CRAY Advanced Technical Workshop in Bologna (June ’04)
BlueGene/Light Overview

- **Node**
  - Dual Core Embedded PowerPC 440
  - 256MB or 512MB memory

- **Network**
  - 3-D torus (point-to-point) & Tree network (broadcast, ...)

- **700MHz, 500MHz prototype (versions of both tested)**

- **4 floating-point per cycle**
  - 2.8 GFlops per processor core

- **Use either 1 PE or 2 PEs per node**

- **Largest system - Lawrence Livermore, 2005 (ASC)**
  - 32 x 32 x 64 nodes (64K nodes, 128K processor cores)
  - Peak performance: 360 Tflops

- **Small physical footprint**
  - 2 nodes per compute card, 16 cards per board, 32 boards per rack
Sweep3D model validation on BG/L

Model prediction error:
- 7.2% (maximum)
- 1.7% (average)

NB: VNM vs. COP mode (using 2 vs 1 PEs per node):
- Factor of ~1.9x higher performance

700MHz
SAGE model validation on BG/L

- **700MHz**
- **Model prediction error:**
  - 10.1% (maximum)
  - 4.1% (average)

- **NB: VNM vs. COP mode (using 2 vs 1 PEs per node):**
  - Factor of ~1.1x higher performance
Relative performance: BG/L to ASCI Q

- 2 regions in graph:
  - equal processor count (up to 8,192 processors)
  - ASCI Q fixed size (above 8,192 processors)

Equal PE count:
BG/L is \(~0.42\times\) speed of Q

Full-sized system:
BG/L is \(~5.5\times\) faster than Q

(5x5x400 sub-grids with best blocking)
Relative Performance: BG/L to ASCI Q

- 2 regions in graph:
  - equal processor count (up to 8,192 processors)
  - ASCI Q fixed size (above 8,192 processors)

Equal PE count:
BG/L is \(\sim 0.5x\) speed of Q

Full-sized system:
BG/L is \(\sim 4.5x\) faster than Q

5x5x400 sub-grids with best-blocking
Expected SAGE Performance (Cray XD1)

- Used measured performance on a 2.0GHz Opteron
- Used measured performance for Uni-directional messages (MPI)
  - Optimistic as Sage mainly used bi-directional

- Predictions for 3-D mesh - results in contention on large PE counts, and for a network topology that matches the application communication pattern
Systems

Two thrusts

A. Efficiency: detection, identification, and elimination of system noise;

B. Fault tolerance: efficient, transparent, system-level incremental checkpointing.
A. Noise detection, identification, and elimination

This is currently a relatively minor effort, though we have a recent success—a ‘little brother’ to “The Case of the Missing Supercomputer Performance.”

What is new here is the source of the system noise and part of the strategy used to identify it.
Analysis of System Overhead

- Demonstration, by the way of a case study of a methodology for analyzing and evaluating the impact of system activity on application performance

- Our methodology has three major components
  1. A set of simple benchmarks
  2. A kernel-level profiling tool, Oprofile to characterize all relevant events and their sources
  3. A Linux 2.6 kernel module that provides timing information for in depth modeling of frequency and duration of each relevant event and determines which sources have the greatest impact on performance (and therefore the most important to eliminate)
Noise Evaluation on an AMD Opteron, Dual Processor

Noise asymmetry between processor 0 and 1
B. Efficient, transparent, system-level incremental checkpointing

Overview

1. Survey: never been done for Linux, or as a set of general-purpose, modular tools;
2. Feasibility: plausibly demonstrated;
3. Implementation: prototype implementation underway.
3. Feasibility

Is current hardware adequate? Must consider

- Bandwidth requirements, a function of processor, memory, I/O bus, disk, and networks speeds, and the behavior of applications.
- This is performance analysis and modeling!
- We have compiled a large number of internal reports summarizing our findings; these represent a significant fraction of our working knowledge.
Characterization

Data Initialization

Regular Processing Bursts

**Sage-1000MB**

- Incremental Ckpt Data (MB)
- Execution time (s)
Our preliminary analysis show that automatic, frequent, and user-transparent incremental checkpointing is a viable technique to provide fault-tolerance for scientific computing.

- The per-process bandwidth slightly decreases as we increase the number of processors (weak scaling).
- Per-process bandwidth is sublinear with the number of processors.
4. Implementation

Implementation of prototype incremental checkpointer is underway.

- Goal is a small set of modular building blocks that could be used in diverse ways;
- **Two components:**
  1. Kernel module for memory transfer;
  2. Loader for executables on NIC (Elan4).

Next

3. Remote storage to disk.

Prototype under development is currently single-node.
Single Processor Performance

- Expanding and refining LANL’s memory model: answering questions about future architectures (multi-core), OS scheduling
- Came up with a diagnostic model for Sage and Partisn that shows the major areas of inefficiencies in single-processor performance
- Compare LANL and Rice single-processor models (role of prefetch instruction will be a joint study)
- Organized a workshop on HPM at HPCA-11 to influence functionality in future architectures (co-organized with Tennessee and Rice).