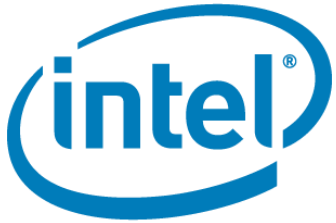


Performance Analysis on Petaflop clusters

Presenter: David Levinthal
Principal Engineer

Business Group, Division: DPD, SSG

Version 1.0
July 28, 2010

Legal Disclaimer

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

  UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

  Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

  The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

  Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.
- All products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.
- Customers, licensees, and other third parties are not authorized by Intel to use Intel code names in advertising, promotion or marketing of any product or service.
- Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations
-

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the first quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be the important factors that could cause actual results to differ materially from the corporation's expectations. Current uncertainty in global economic conditions pose a risk to the overall economy as consumers and businesses may defer purchases in response to tighter credit and negative financial news, which could negatively affect product demand and other related matters. Consequently, demand could be different from Intel's expectations due to factors including changes in business and economic conditions, including conditions in the credit market that could affect consumer confidence; customer acceptance of Intel's and competitors' products; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Intel operates in intensely competitive industries that are characterized by a high percentage of costs that are fixed or difficult to reduce in the short term and product demand that is highly variable and difficult to forecast. Revenue and the gross margin percentage are affected by the timing of new Intel product introductions and the demand for and market acceptance of Intel's products; actions taken by Intel's competitors, including product offerings and introductions, marketing programs and pricing pressures and Intel's response to such actions; Intel's ability to respond quickly to technological developments and to incorporate new features into its products; and the availability of sufficient supply of components from suppliers to meet demand. The gross margin percentage could vary significantly from expectations based on changes in revenue levels; capacity utilization; excess or obsolete inventory; product mix and pricing; variations in inventory valuation, including variations related to the timing of qualifying products for sale; manufacturing yields; changes in unit costs; impairments of long-lived assets, including manufacturing, assembly/test and intangible assets; and the timing and execution of the manufacturing ramp and associated costs, including start-up costs. Expenses, particularly certain marketing and compensation expenses, as well as restructuring and asset impairment charges, vary depending on the level of demand for Intel's products and the level of revenue and profits. The recent financial crisis affecting the banking system and financial markets and the going concern threats to investment banks and other financial institutions have resulted in a tightening in the credit markets, a reduced level of liquidity in many financial markets, and extreme volatility in fixed income, credit and equity markets. There could be a number of follow-on effects from the credit crisis on Intel's business, including insolvency of key suppliers resulting in product delays; inability of customers to obtain credit to finance purchases of our products and/or customer insolvencies; counterparty failures negatively impacting our treasury operations; increased expense or inability to obtain short-term financing of Intel's operations from the issuance of commercial paper; and increased impairments from the inability of investee companies to obtain financing. Intel's results could be impacted by adverse economic, social, political and physical/infrastructure conditions in the countries in which Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Intel's results could be affected by adverse effects associated with product defects and errata (deviations from published specifications), and by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust and other issues, such as the litigation and regulatory matters described in Intel's SEC reports.

Petaflop scale analysis methodology

- Most slides taken from https://openlab-mu-internal.web.cern.ch/openlab-mu-internal/00_News/News_pages/2010/10-15_Meeting_David_Levinthal/10-15_Meeting_David_Levinthal.htm
  - 60 slide presentation on Xeon 5500 microarchitecture
    - Introduction
  - 190+ slide presentation on HW event performance analysis/sw optimization
    - What would I do with 100 million lines of code when told to just make it go fast

Basic performance analysis and SW optimization methodology

- Tune the machine
  - Make sure bios, dimms, disks etc are correctly configured
- Tune the compiler usage
  - only optimize what uses time
- Remove the pipeline stalls
- Remove the single thread resource limitations
- **Improve the scaling**

A Hypothetical Petaflop Cluster

- Not based on any existing or future planned processor known to the author
- Assumptions
  - 24 DP Gigaflops per core
    - 3 Ghz/core – and AVX instruction set
  - 1.2 Teraflops/box
    - 50 cores
  - 850 boxes

Note: many of these slides taken from:

Scaling on large clusters of multicores

- Large cluster MPI job scaling tends to be limited by all-to-all MPI APIs
  - N*Log(N) or even N! (if really badly coded ☺)
- Experience has shown that using shared memory threading within the box can be used to reduce the MPI node count substantially.
- The result of such a combination is that the node count is not largely higher than today's clusters, but multi thread scaling must be improved.

Event Classes: High Level View

1. Execution flow events
   - Cycles, Branches, stalls, uops/inst_retired
   - Guide compiler usage
2. Penalty events (ONLY useful if they have well defined cost)
   - Ex: load requiring access to dram
   - Modify code/build to reduce penalties/pipeline stalls
3. Resource saturation events
   - Bandwidth, load/store buffers, dispatch ports
   - No well defined cost
   - Change data layout/access patterns
4. Architectural characterization
   - Cache accesses, MESI states, snoops
   - Used to improve silicon design, not application performance
5. Instruction mix
   - Do not measure what you think, extremely difficult to validate

Less than ideal multi core scaling

- Perfect scaling results in the number of perf events (summed over cores) being constant
- Difference of event counts can identify locality using cycles and some reasons for non scaling behavior
  - Cacheline access contention can cause non scaling
    - Load-hitm and store address analysis identifies this
- Most non scaling due to resource saturation and evaluated as a ratio: events/wall_cycles
  - Wall_cycles ~ cycles/active cores
    or Cpu_clk_unhalted.thread max(ICPU)
  - **Cannot be seen in difference display**

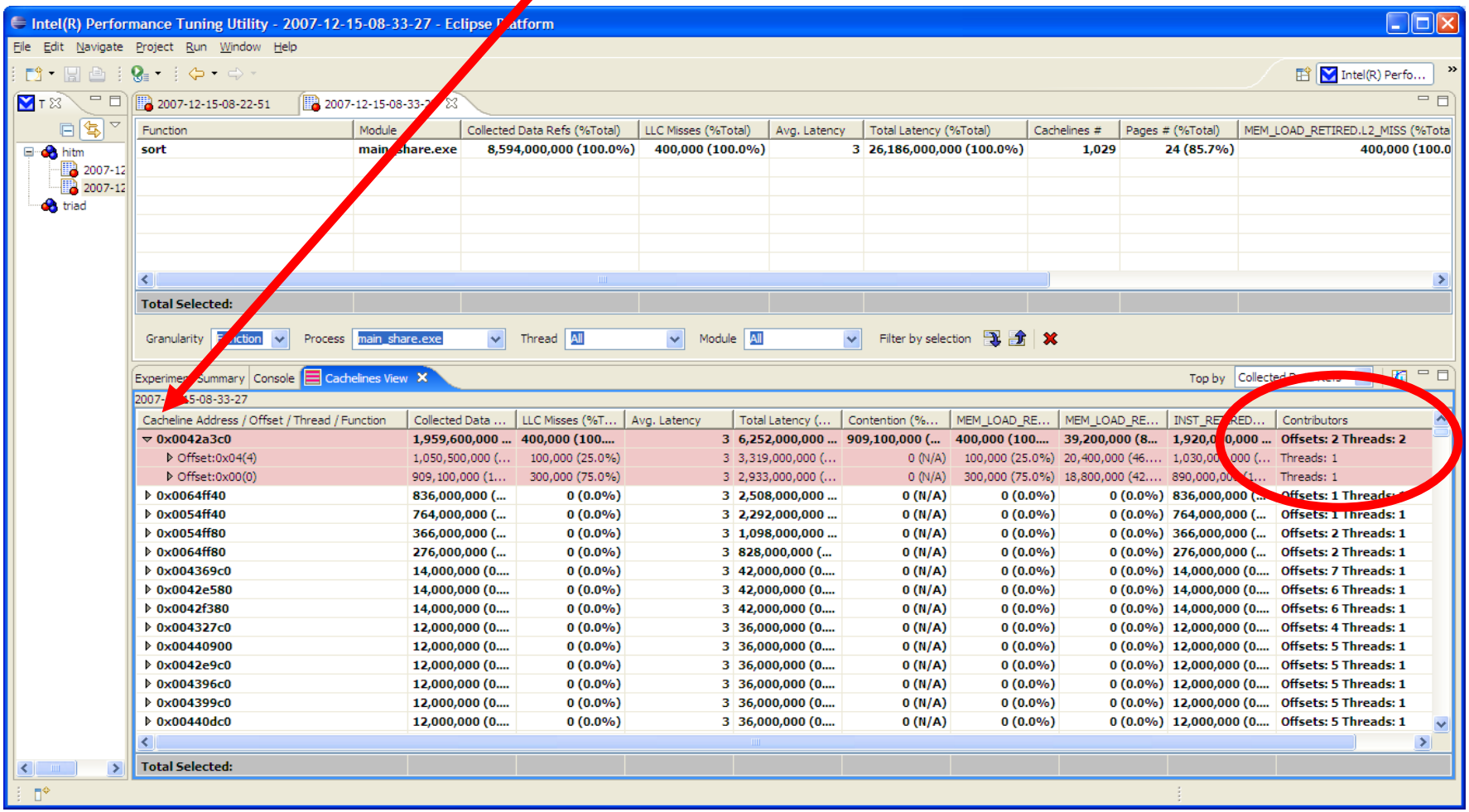
Superfluous sources/signatures of non scaling

- Turbo
  - Having this on results in large drop from 1->2
- Smaller share of LLC
  - Decrease in LLC hits, increase in LLC miss
- Increase in page faults
  - More threads require more memory
- Asymmetry associated with core 0
  - OS induced imbalance
- Context switching
  - OS's love to move things around, being the boss!
  - Don't know about logical cores & double up on one physical core, while other phys cores are idle

Sources/signatures of non scaling

- Saturating a resource
  - Ex: Bandwidth
  - Code optimization increases resource saturation
- Shared memory application specific
  - Serial execution
  - Overly contested lock access
  - False sharing (non overlapping access to a line)
- NUMA based non scaling
  - Increase in *.remote_dram

Expanding the "arrow" we see the 2 threads access the line at Different Offsets…This is False Sharing



More sources of non scaling

- Load imbalance
  - Increase in halted cycles
- MPI global operations
  - increase in time associates with MPI global APIs
    - Ex: allreduce
- Synchronous message passing
  - "Intrinsically" non scaling
- HT can be viewed as a way to recover scaling

Dominant issues

- Within the box
  - Thread interactions (hitm/store analysis)
  - Serial portion
  - Threading decomposition load imbalance
- Over the cluster
  - MPI all to all
  - Load imbalance

Resolving non scaling issues

- Disable turbo while doing measurements
- Disable HT while doing measurements
- Pin all affinities
  - OS's love to move things
  - Old OS's will schedule 2 threads on a physical core while leaving other physical cores idle. This increases with thread count
- Make sure there is enough memory
  - /proc/meminfo->Active (?)
- Do 1 thread baseline on a core other than 0
- Increased LLC miss
  - Usual approaches to fixing these, see large talk

Resolving non scaling issues

- Bandwidth issues
  - Check data decomposition for separation
  - Improve data layout to reduce cacheline consumption
  - See previous section on BW issues
- Excessive lock contention
  - Use finer grained locking
  - Use faster locking APIs
  - Make sure the global update is really needed
    - Can you continue working with local copy
- False sharing
  - Put 64 bytes between data elements

Resolving non scaling issues

- NUMA related non scaling
  - Remote dram data access
    - Improve buffer initialization for local access
    - Make multiple copies for each socket
  - Remote dram ifetch access
    - Make two binaries on the disk and affinity pin per socket
- MPI global operations
  - Use OpenMP (Cilk, TBB, Pthreads) within a box to reduce MPI nodes
  - Use good MPI library

Resolving non scaling issues

- Load imbalance
  - Seen as halted cycles
    - TSC difference for successive cpu_clk_unhalted.ref != SAV
  - Work queue approach dynamically restores balance
    - At a cost
      - NUMA locality can be lost
      - SW prefetching can become unpredictable within a thread
  - Estimate work during data decomposition to create balanced work rather than balanced iteration count
  - Save some iterations for final work queue balancing

Summary

- Petaflop scaling problem can be reduced to single box thread scaling for many issues
- Event based sampling performance analysis is extremely powerful
- Correct methodology is essential
- Correct usage of events is essential