Oversubscription on Multicore Processors

Costin Iancu, Steven Hofmeyr, Filip Blagojević, Yili Zheng

Lawrence Berkeley National Laboratory

Parallel & Distributed Processing (IPDPS), 2010

Motivation

- Increasingly parallel and asymmetric hardware (architecture + performance)
- Existing runtimes in competitive environments
- Partitioning vs. sharing on real hardware

Oversubscription

+

- Compensate for data and control dependencies
- Decrease resource contention
- Improve CPU utilization

−

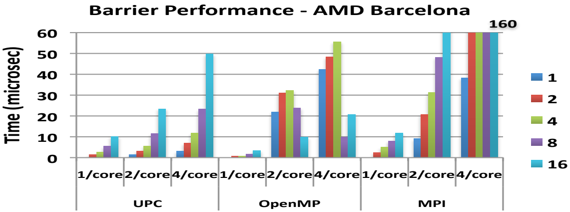
- Overhead for migration, context switching and lost hardware state (*negligible*)
- Slower synchronization due to increased contention

Setup

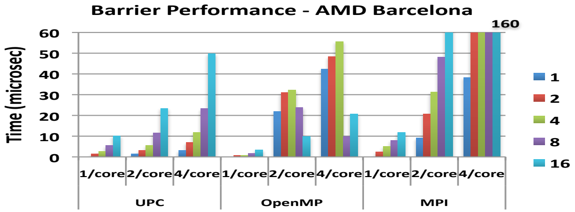
- MPI (MPICH 2), UPC, OpenMP
- Synchronization: poll + yield
- Linux 2.6.27, 2.6.28, 2.6.30
- Intel compiler with −O3
- NPB without load imbalances (separate paper)

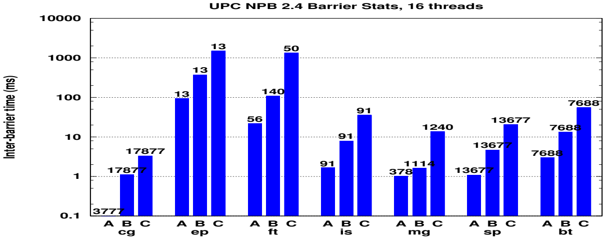
	Processor	Clock GHz	Cores	L1 data/instr	L2 cache	L3 cache	Memory/core	NUMA
Tigerton	Intel Xeon E7310	1.6	16 (4x4)	32K/32K	4M / 2 cores	none	2GB	no
Barcelona	AMD Opteron 8350	2	16 (4x4)	64K/64K	512K / core	2M / socket	4GB	socket
Nehalem	Intel Xeon E5530	2.4	16 (2x4x2)	32K/32K	256K / core	8M / socket	1.5G / core	socket

Benchmark Characteristics

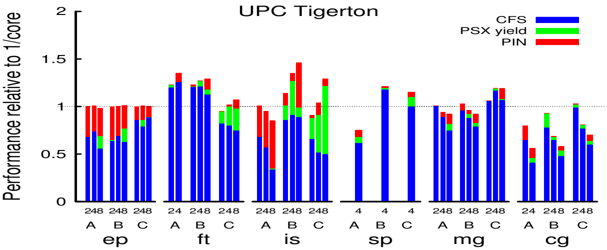


Benchmark Characteristics



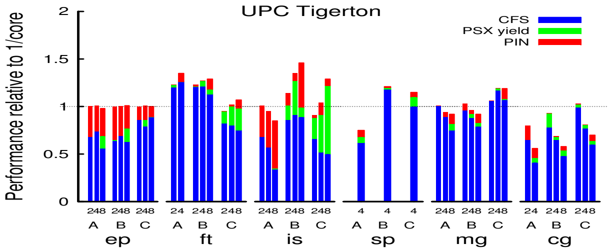


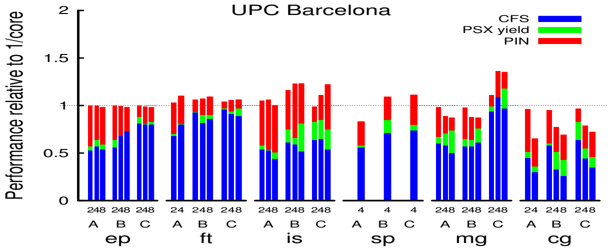
UPC — UMA vs. NUMA



- sched_yield: default vs. POSIX
- Pinning affects variance (120 % vs. 10 %) and memory affinity

UPC — UMA vs. NUMA





- sched_yield: default vs. POSIX
- Pinning affects variance ($120\,\%$ vs. $10\,\%$) and memory affinity

- Small overall effect ($\pm\ 2\,\%$ avg)
- EP: computationally intensive
- FT, IS: improvement up to $46\,\%$
- SP, MG: problem size $\leftrightarrow$ granularity
- CG: degradation up to $44\,\%$

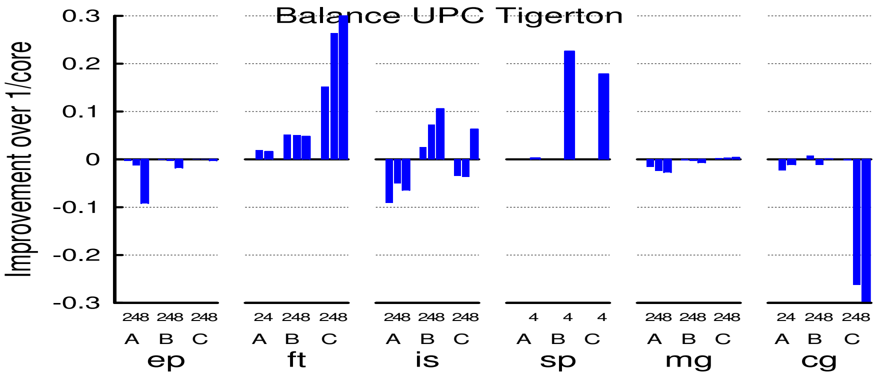


Figure 5. *Changes in balance on UMA, reported as the ratio between the lowest and highest user time across all cores compared to the 1/core setting.*

Cache Miss Rate (LLC / L2)

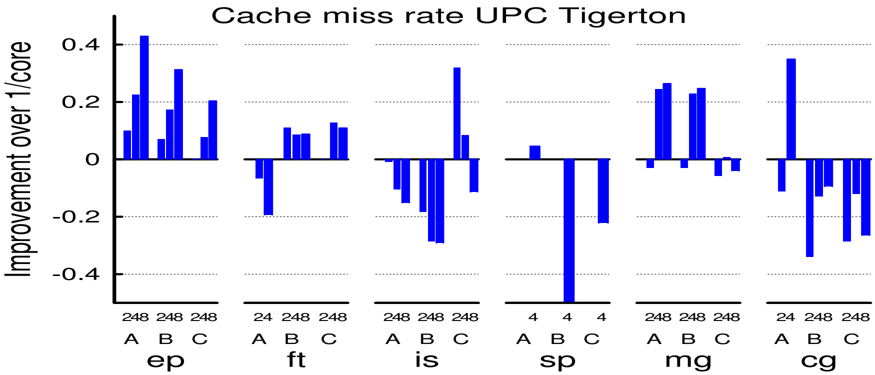
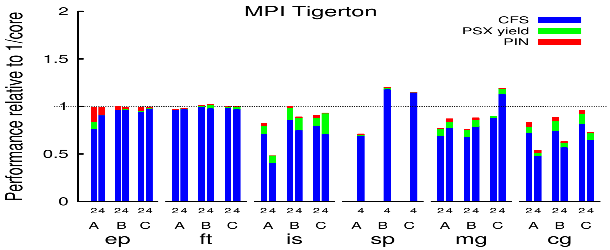


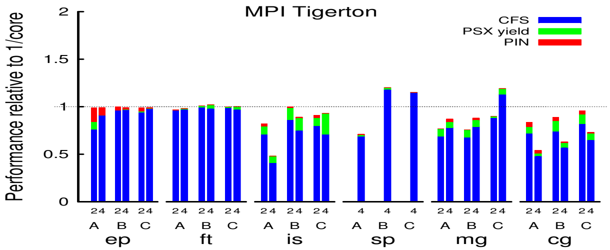
Figure 6. *Changes in the total number of cache misses per 1000 instructions, across all cores compared to 1/core. The EP miss rate is very low.*

MPI and OpenMP

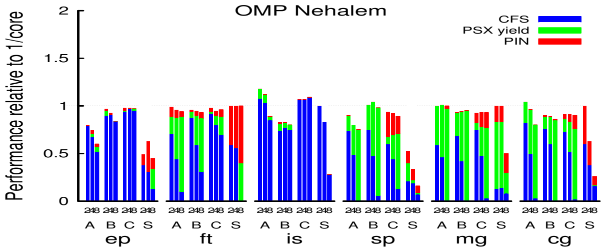


- Overall decrease by $10\,\%$
- Caused by barrier overhead (cp. modified UPC)

MPI and OpenMP



- Overall decrease by $10\,\%$
- Caused by barrier overhead (cp. modified UPC)



- Slight degradation
- Best performance with OMP_STATIC
- KMP_BLOCKTIME
  - 0 Improvement up to $10\,\%$ for fine-grained benchmarks
  - ∞ Best overall performance

Competitive Environments

- Sharing (best effort) vs. Partitioning (isolated on sockets)
- One thread per core
  - Overall $33\,\%/23\,\%$ improvement with sharing for UPC/OpenMP on Barcelona (CMP) but no difference for Nehalem (SMT)
  - Better for application with differing behavior
- Oversubscription ...
  - improves benefits of sharing for CMP
  - changes relative order of performance for UPC, MPI, OpenMP
- Imbalanced sharing possible

Conclusion

"Intuitively, oversubscription increases diversity in the system and decreases the potential for resource conflicts."

"All of our results and analysis indicate that the best predictor of application behavior when oversubscribing is the average inter-barrier interval. Applications with barriers executed every few ms are affected, while coarser grained applications are oblivious or their performance improves."

"We expect the benefits of oversubscription to be even more pronounced for irregular applications that suffer from load imbalance."