Bandwidth Bandit: Understanding Memory Contention

UPMARC Uppsala Programming for Multicore Architectures Research Center

David Eklov, Nikos Nikoleris, David Black-Schaffer and Erik Hagersten

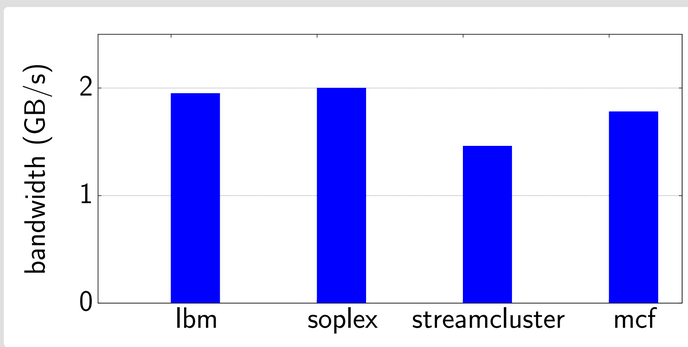
UPPSALA UNIVERSITET

Motivation

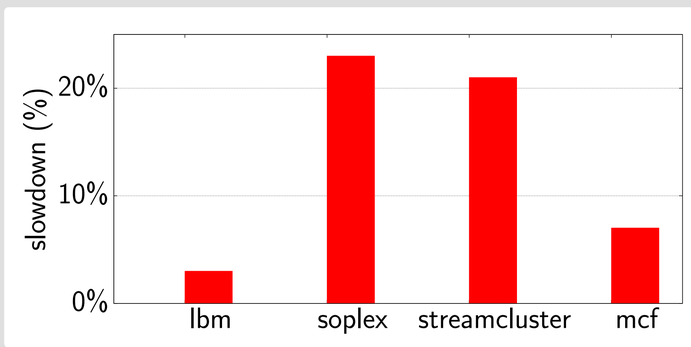
Contention for off-chip memory bandwidth is increasingly important:

1. can have large impact on application performance [1, 3] and
2. is likely to increase in the future [2].

. . . but bandwidth demand != sensitivity



Applications with similar bandwidth demands

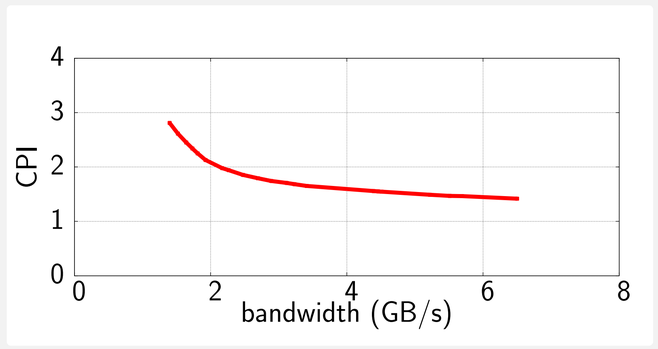


have different slowdowns due to contention for the off-chip memory.

Goal: Analyze Memory Contention.

To understand the impact of memory contention we need:

CPI=f(BW)



Quantitative data that allows us to analyze the impact of memory contention.

Bandwidth Bandit

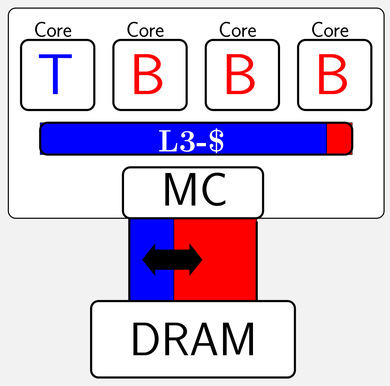
Profiling tool for measuring applications' sensitivity to memory contention.

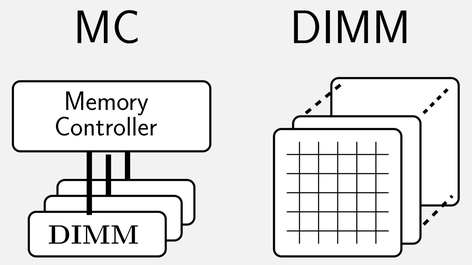
Works as follows:

- Co-runs the *Target* application with a *Bandit* application
- The Bandit "steals" memory bandwidth from the Target
- Varies the amount of bandwidth stolen while measuring the Target

Result:

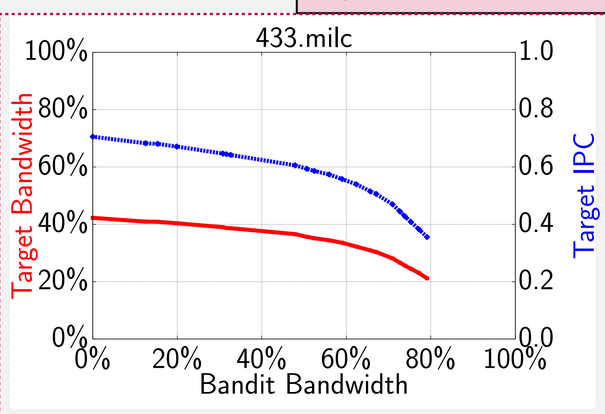
- Target's IPC as a function of its available memory bandwidth

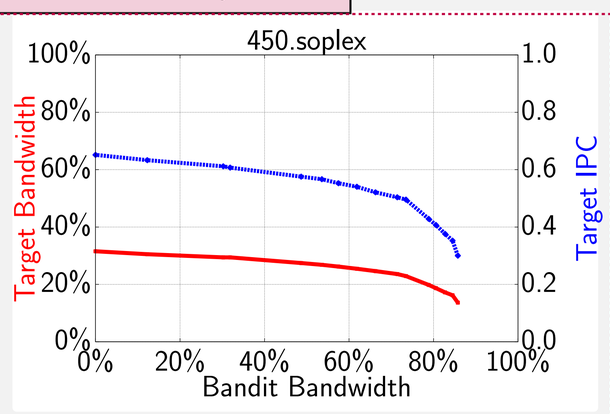




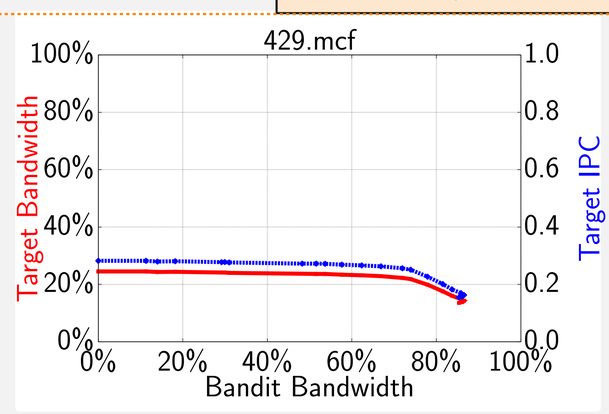
Results

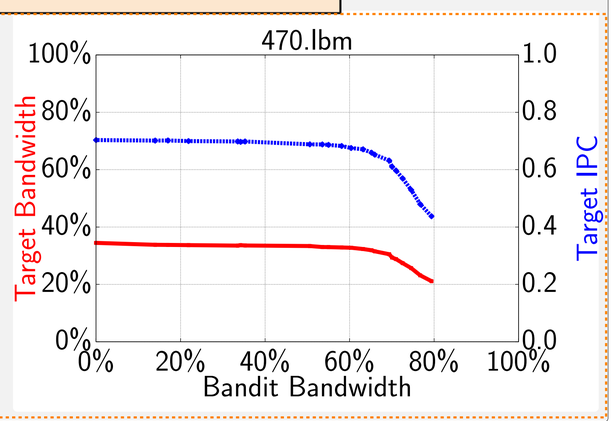
large slowdowns before bw saturates → latency sensitive





slowdowns only when bw saturates → bandwidth sensitive

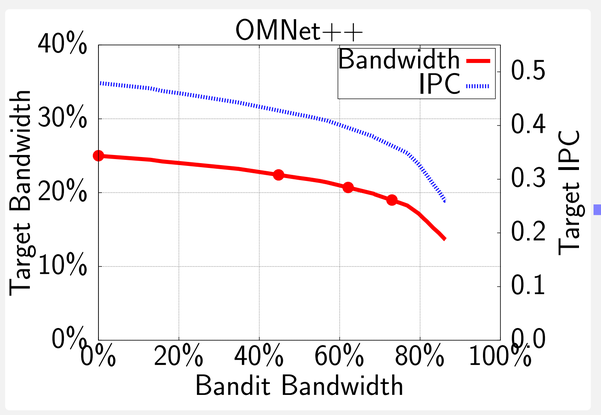


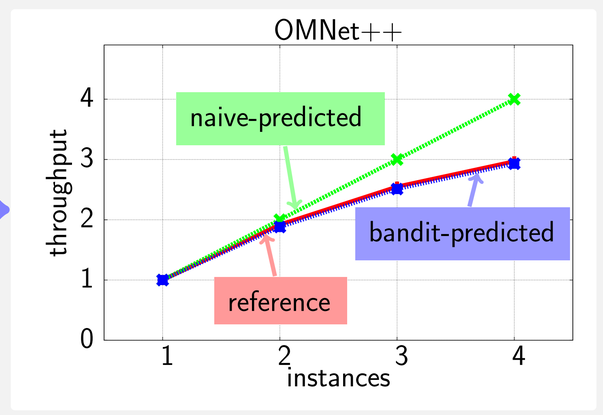


Case Study

Predict the performance impact of memory contention when co-running one, two, three and four instances of OMNet++.

- Reference
  1. Co-run all instances and measure the aggregate throughput
- "Naive" Prediction
  1. Assume that there is no slowdown as long as the instances' total bandwidth is less than the systems peak bandwidth.
- Bandit Graphs
  1. Use bandwidth graphs to estimate instances' bandwidth
  2. Then, use their bandwidth to estimate their IPCs



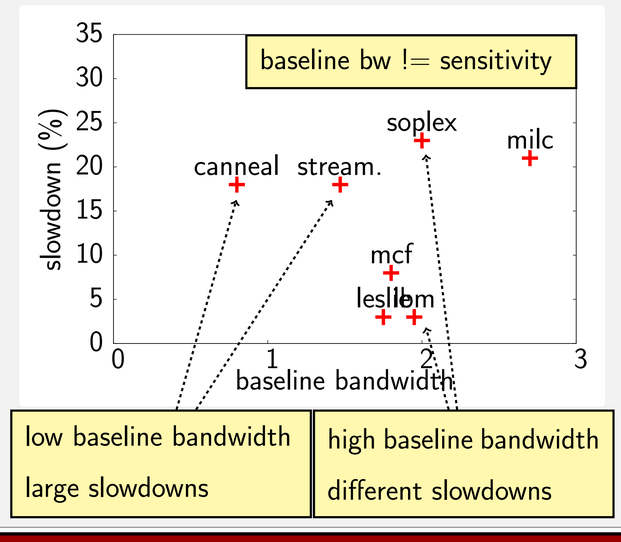


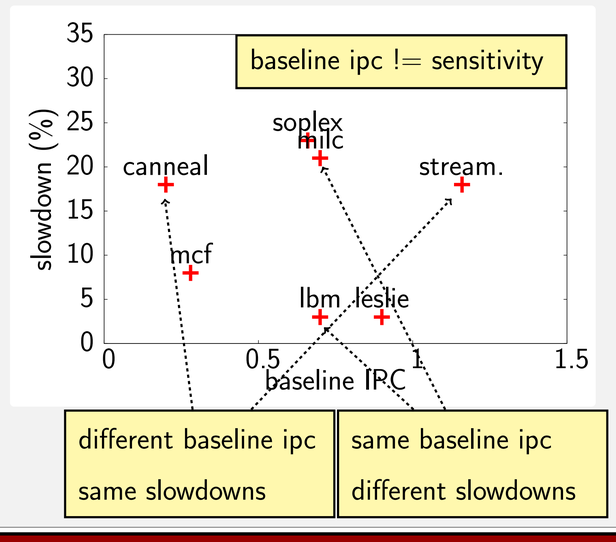
Result: The prediction based on Bandwidth Bandit data almost perfectly matches the reference throughput.

Sensitivity to memory contention

Contrary to previous results [1, 3] neither the *baseline bandwidth* nor the *baseline IPC* are good indicators of an application's sensitivity to memory contention.

- Baseline Bandwidth – Application's bandwidth when running alone
- Baseline IPC – Application's IPC when running alone
- Slowdown – Baseline IPC / IPC at 90% of saturation bandwidth





References

[1] T. Dey, W. Wang, J. W. Davidson, and M. L. Soffa.
Characterizing multi-threaded applications based on shared-resource contention.
In *Proc. of ISPASS*, 2011.

[2] B. M. Rogers, A. Krishna, G. B. Bell, K. Vu, X. Jiang, and Y. Solihin.
Scaling the bandwidth wall: challenges in and avenues for CMP scaling.
In *Proc. of ISCA*, 2009.

[3] L. Tang, J. Mars, N. Vachharajani, R. Hundt, and M. L. Soffa.
The impact of memory subsystem resource sharing on datacenter applications.
In *Proc. of ISCA*, 2011.