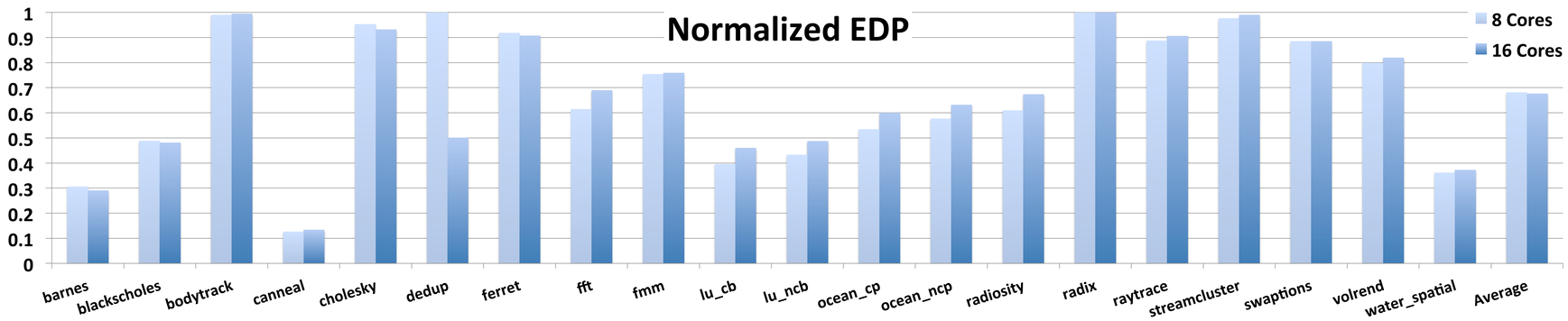
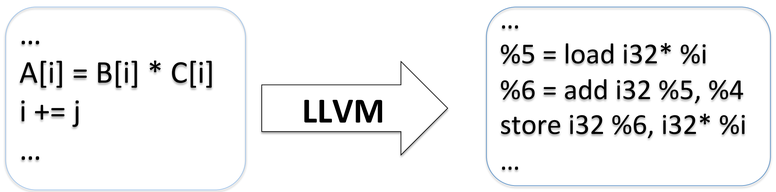
Dynamically Reconfiguring Multi-Core Architectures using Task Graph based Analysis

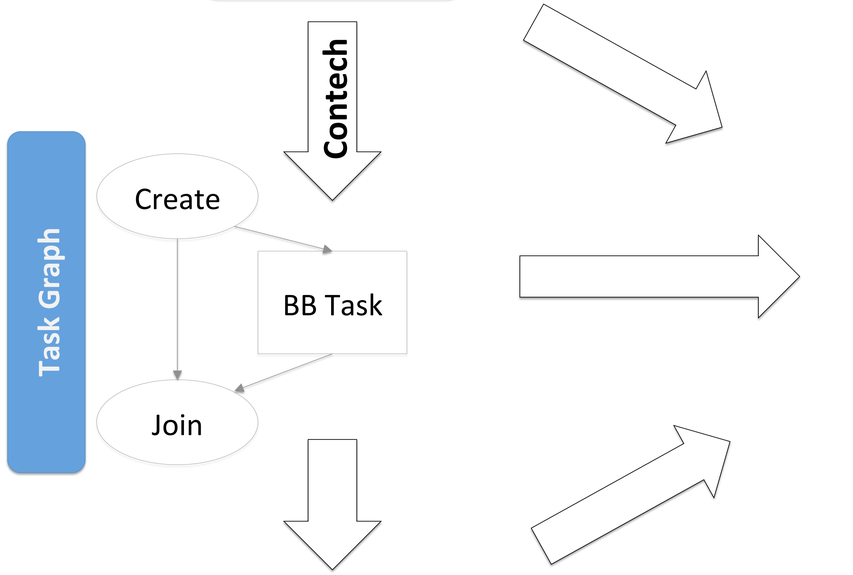
Sriseshan Srikanth, Brian P. Railing, Thomas M. Conte
School of Computer Science, Georgia Institute of Technology

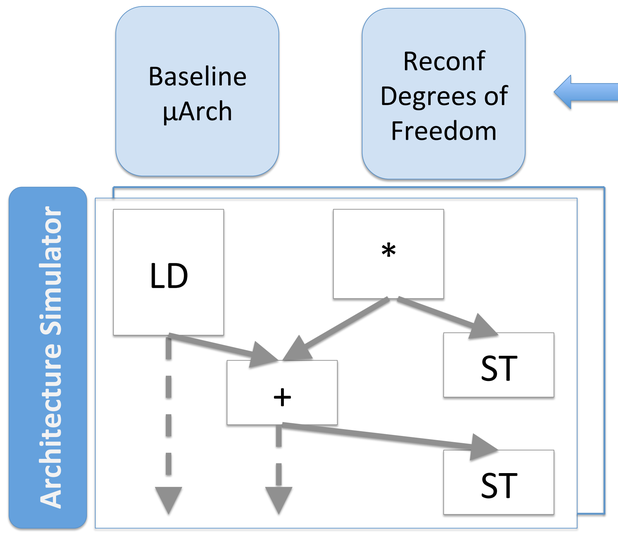
- There is significant variation in resource usage and requirements across generic parallel workloads, across threads within a workload as well as across various phases within a thread.
- Significant performance and energy benefits result from dynamically reconfiguring multi-core architectures to achieve optimal resource utilization.





A hardware agnostic dynamic trace of a parallel program complete with basic block information, memory address and synchronization traces.



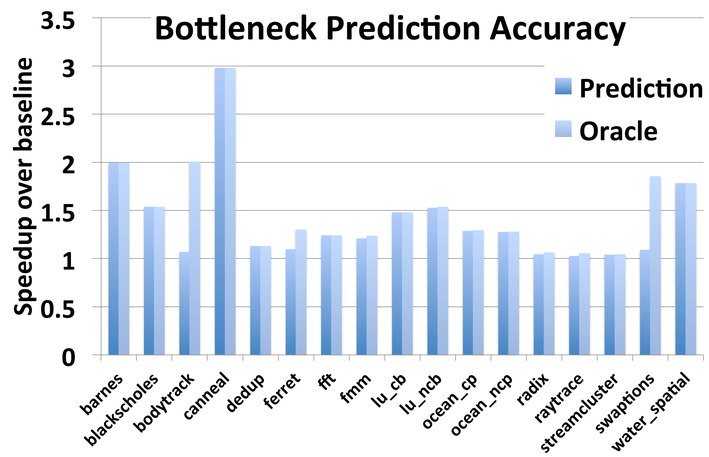


Potentially double the count of a single bottlenecked function unit per task.

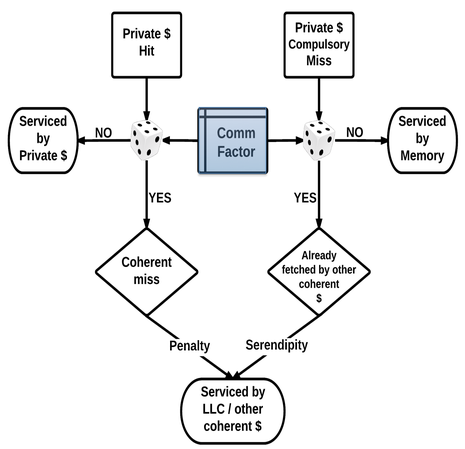
Average normalized energy-delay product reduction of 32%

Function Unit	Baseline Count	Function Unit	Baseline Count
Int Add	3	FP Add	1
Int Mul	1	FP Mul	1
Int Div	1	FP Div	1

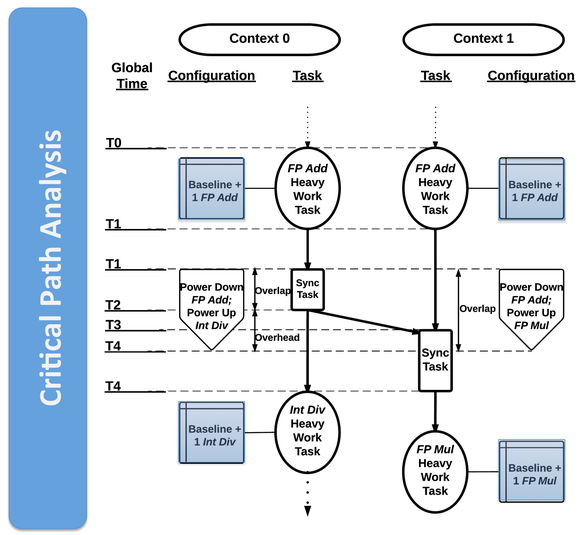
The plot below depicts the accuracy of the predicted bottleneck resource vis-à-vis an oracle dynamic configuration obtained by a brute force configuration sweep.



An analysis that abstracts coherence misses into a stochastic variable.



An analysis that computes critical path of a Task Graph using the recorded task times and speedup estimates.



References

- Brian P. Railing, Eric R. Hein, and Thomas M. Conte. 2015. Contech: Efficiently Generating Dynamic Task Graphs for Arbitrary Parallel Programs. ACM Trans. Archit. Code Optim. 12, 2, Article 25 (July 2015), 24 pages.
- V. Cabezas and M. Puschel. "Extending the roofline model: Bottleneck analysis with microarchitectural constraints". In: Workload Characterization (IISWC), 2014 IEEE International Symposium on. 2014, pp. 222–231