

# iFKO, LANB, PWML, PCA & Other Fascinating Post-ICL Acronyms

R. Clint Whaley

(whaley@cs.utsa.edu)

Dave Whalley

Florida State University

[www.cs.utsa.edu/~whaley](http://www.cs.utsa.edu/~whaley)

Anthony M. Castaldo

(castaldo@cs.utsa.edu)

University of Texas at San Antonio  
Department of Computer Science

- **1991-2001:** ICL/UTK!
- **2002-2004:**  
FSU/PhD/iFKO
  - ① SP&E05: L3 Packed & dense BLAS
  - ② **ICPP05:** iFKO
- **2005-Present:** UTSA
  - ① SIGPLAN SoLCSD07: Qing Yi-POET
  - ② SISC08 : error reduction
  - ③ SP&E08: timer design
  - ④ **CANA08** : LANB
  - ⑤ **ICPP09** : ML
  - ⑥ **PPoPP10:** PCA
  - ⑦ L2BLAS, mem-bound opt
  - ⑧ Rewrite of searches

## What it is

- FKO highly optimized backend (part x86)
- Search scripts to tune all parameters
- Roughly 8 repeatable opts
- Roughly 6 empirically tuned opts
- Enough front-end support for Level 1 BLAS

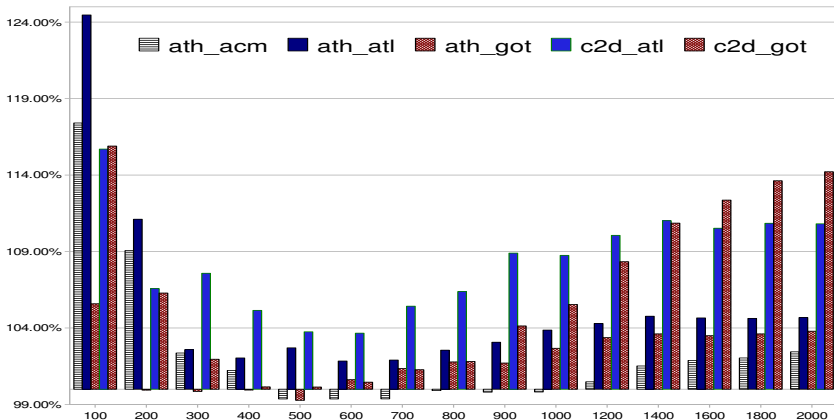
## What we found

- Mem-bound ops must be tuned separately for each cache level
  - Ops tuned for various cache lvls use diff opts & params
  - Provided best overall L1BLAS
- ⇒ Finally understood why compilers haven't made ATLAS obsolete

Why don't compilers make library production obsolete?

## Four anti-HPC Compiler Traditions

- 1 My assumptions trump your experimental results
  - Libraries eventually have users wt. applications
    - keeps them honest to some degree
- 2 All problems solved 20 years ago → nothing works today
  - HPC weak, but does reward raw performance improvement
  - We haven't solved this prob in serial:
    - ⇒ Let's solve it on heterogeneous massively parallel machine!
- 3 Benchmark much more important than application
- 4 10,000 front-ends, 0 HPC backends
  - CISC compaction, front-end (arch) optimization, inst alignment, inst selection & sched

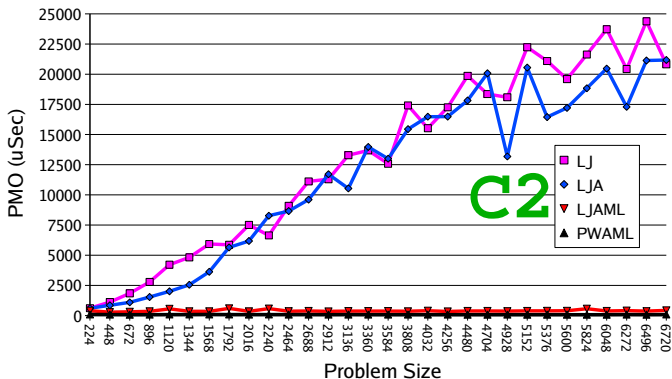


- <http://www.cs.utsa.edu/~whaley/papers/lanb.pdf>
- “Empirically Tuning LAPACK’s Blocking Factor for Increased Performance”, by R. Clint Whaley. *International Multiconference on Computer Science and Information Technology*, Wisla, Poland, Oct 20-22, 2008.

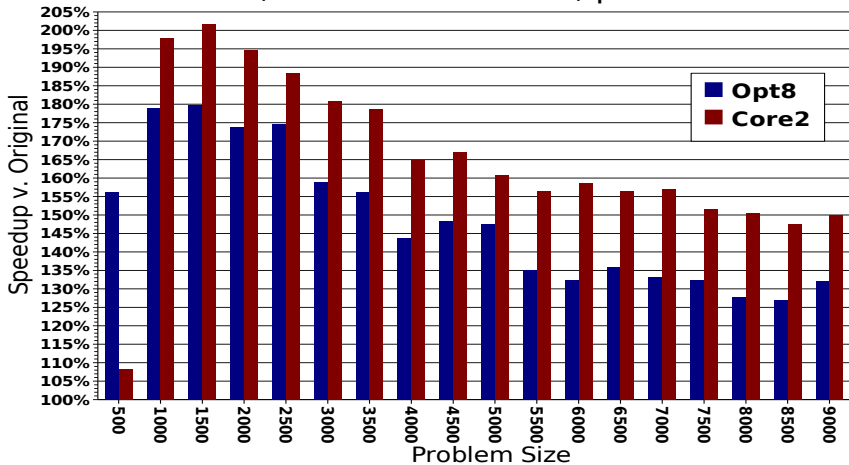
### Paper

Anthony M. Castaldo and R. Clint Whaley, "Minimizing Startup Costs for Performance-Critical Threading" IPDPS2009, pages 1-8, Rome, Italy, May 25-29, 2009.

### LJ, LJA vs. LJAML, PWAML



### LU, New thread vs old, p=8



## Paper

Anthony M. Castaldo and R. Clint Whaley, "Scaling LAPACK Panel Operations Using Parallel Cache Assignment", In *Proceedings of the 2010 ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 223-231, Bangalore, India, January 9-14, 2010.

## Key Points

- $O(N \times NB)$  flops; L2BLAS bus-bound
  - Recur until panel fits in union of parallel caches
  - Split problem by rows, moving parallel overhead into loop
  - Use cache coherence to get hardware-speed thread syncs
  - Use cache-tuned L2BLAS
- ⇒ Achieve superlinear speedup (cache speed not mem)

## Ongoing work

- Full QR support for ATLAS – Siju Samuel
- PCA on GPUs – Kyung Min Su
- 2-sided factorizations – Tony Castaldo
- How to optimize bus-bound operations – Me
- Rewrite of ATLAS's search & tuning infrastructure – Me

## Where to find papers

- <http://www.cs.utsa.edu/~whaley/papers.html>