COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Finding and Optimizing Phases in Parallel Programs

Jeffrey K. Hollingsworth <hollings@cs.umd.edu>
Ray Chen <rchen@cs.umd.edu>

Phases of UMD CS

Computer & Space Sciences: 1962-1987

AV Williams: 1987-2018

Iribe Center: 2018-

- ***CS@UMD: Future is Exciting***
  - *Largest Major on campus (over 2,800 undergrads, plus 400+ Computer Engineering)*
  - *New Building in 2018*
  - *Hiring O(10) New Faculty in couple of years*
  - *New Big Data Masters & Certificate Programs*

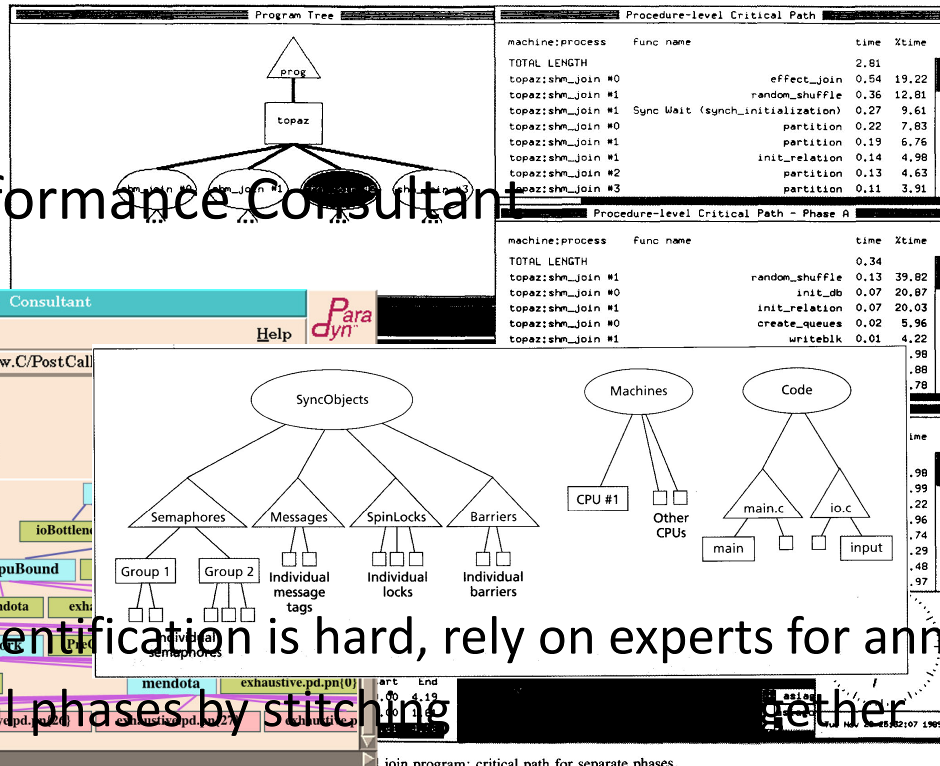
Motivation

- HPC programs often contain “phases”
  - Dynamic execution context
  - Each have distinct performance traits
- Particularly problematic if inside a time-step loop
  - Short phases confound tools
  - Difficult to analyze a rapidly changing landscape
  - Worse if phases are nested

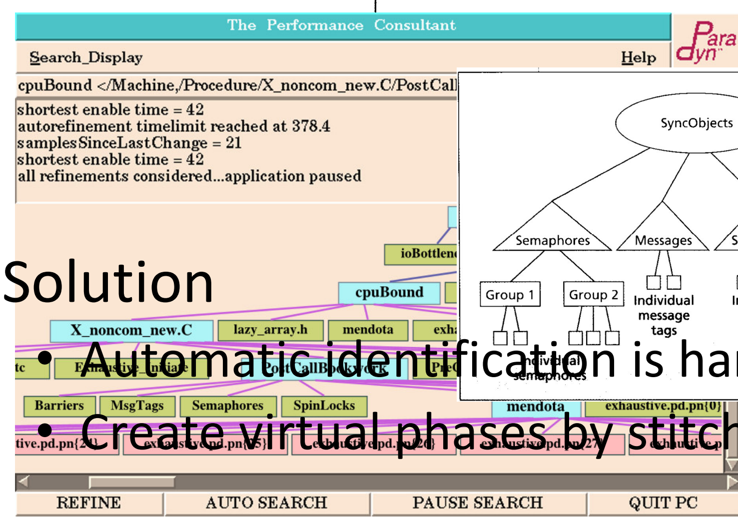
LULESH2 MPI Call Trace

Automatic Phase Identification

- My Failed Prior Attempts
  - IPS-2 (c. 1990)



  - Paradyn's Performance Consultant (c. 1995)



  - Solution
    - Automatic identification is hard, rely on experts for annotations
    - Create virtual phases by stitching together

Guided Phase Identification

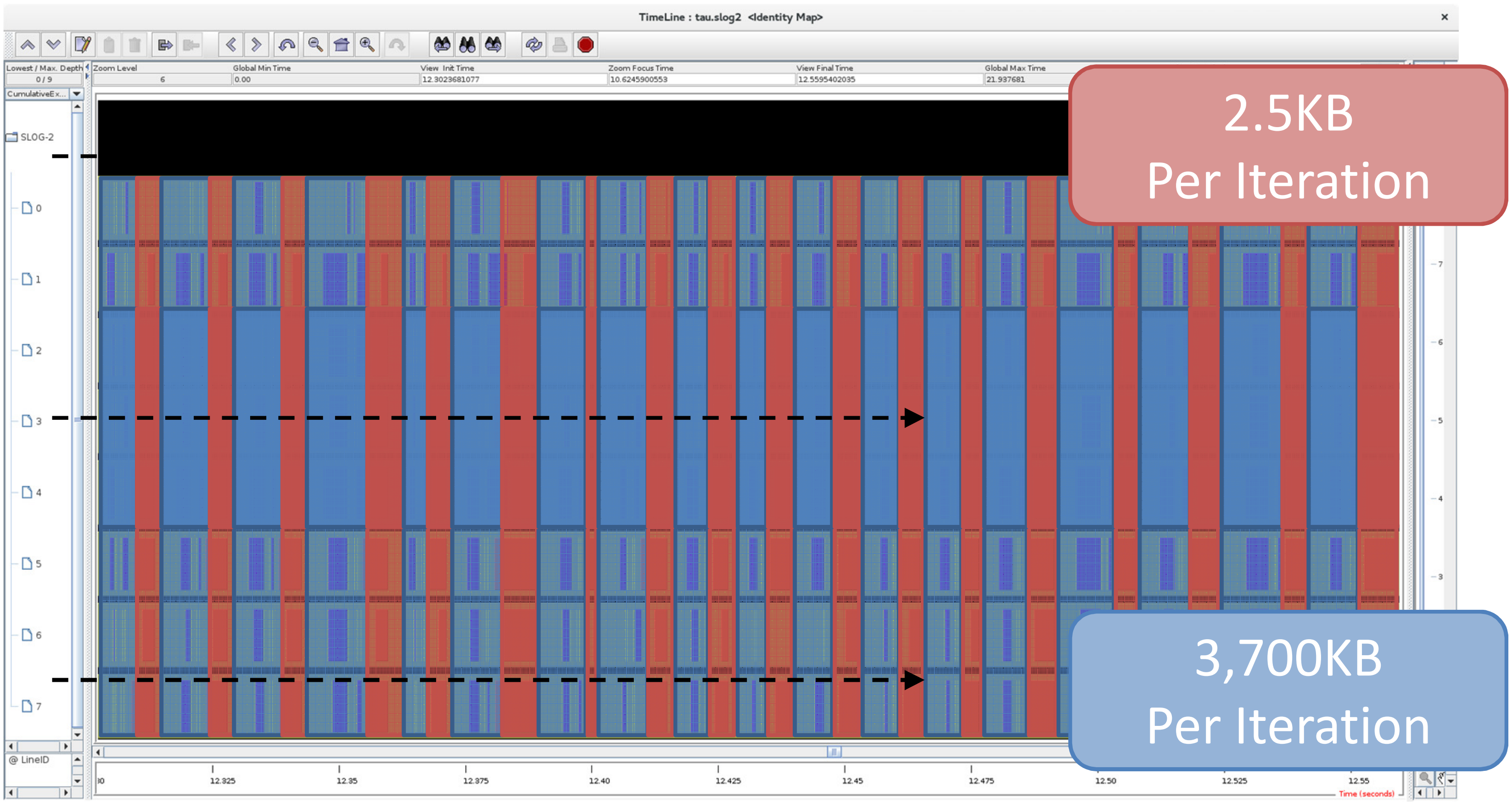
```
while (locDom->time() < locDom->stoptime())
{
    cali::Annotation region1("tuner.communication").begin();
    TimeIncrement(*locDom);
    region1.end();

    cali::Annotation region2("tuner.computation").begin();
    LagrangeLeapFrog(*locDom);
    region2.end()
}
```

Performance Landscape

Contextual Timeline

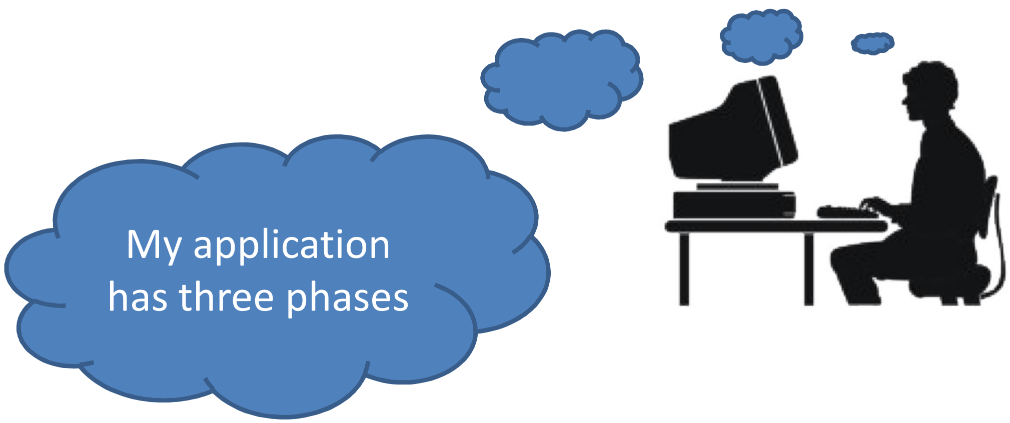
Actual Timeline

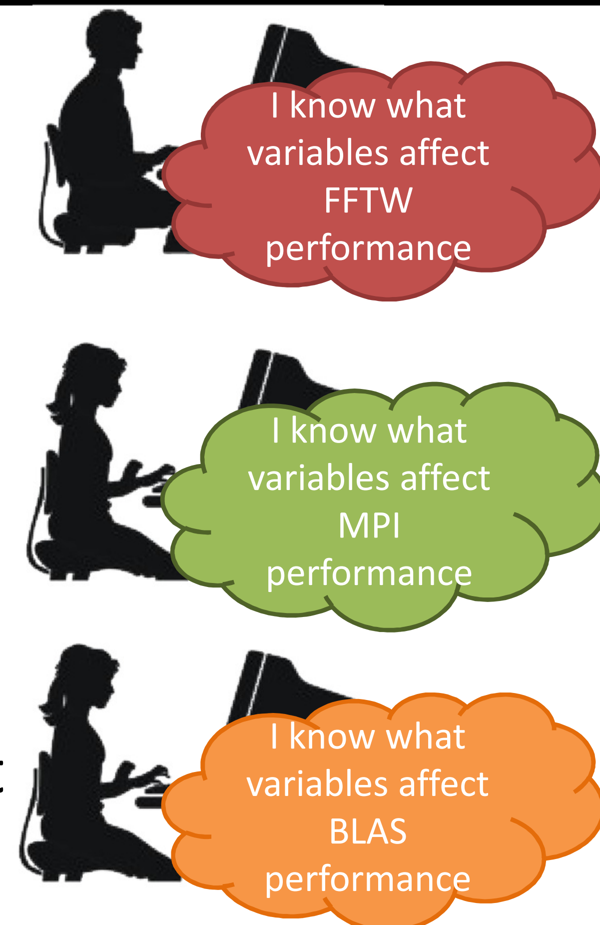
Contextual Timeline



Cross-Domain Analysis

- Utilize experts during development
  - Library writers specify tuning variables
  - Application writers specify code regions





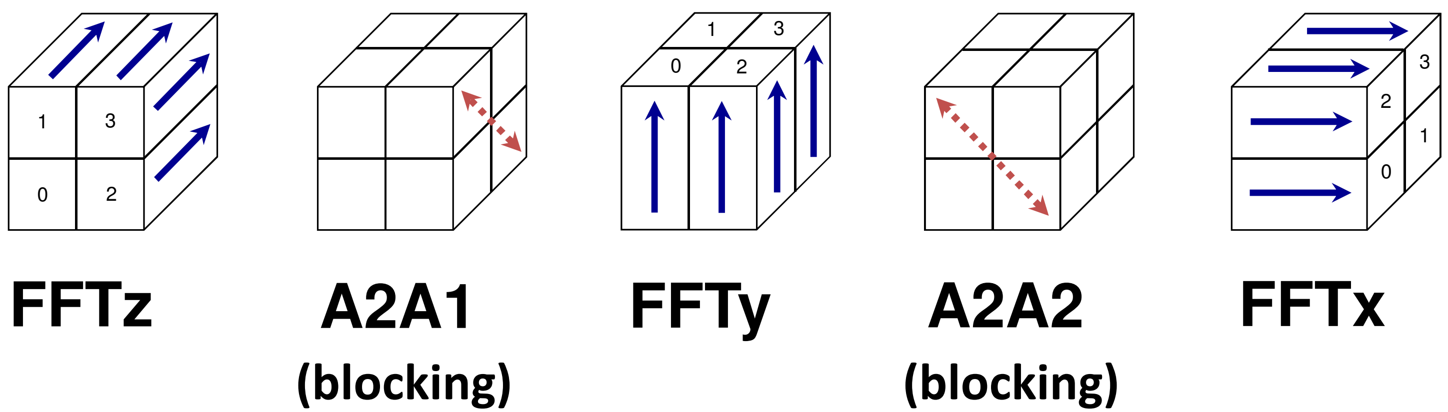
  - Phase dictates different performance context
    - Even though the same function is being called

Integration Work

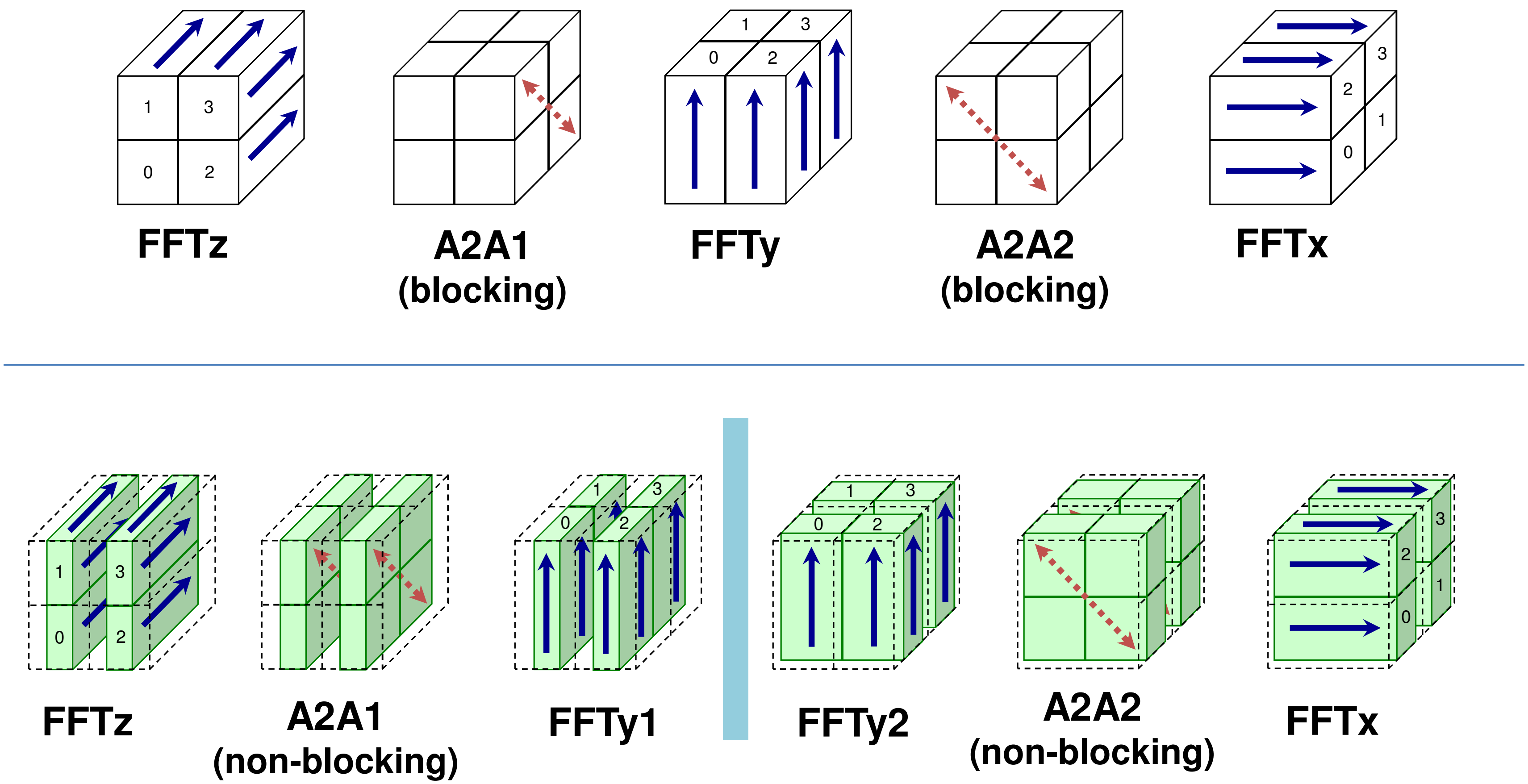
- Special annotation types identify:
  - Tunable variables
  - Code regions that should enable tuning
- New Caliper tuning service
  - Listens for and reacts to special annotations
  - Calls Active Harmony to perform search

3D Fast Fourier Transform

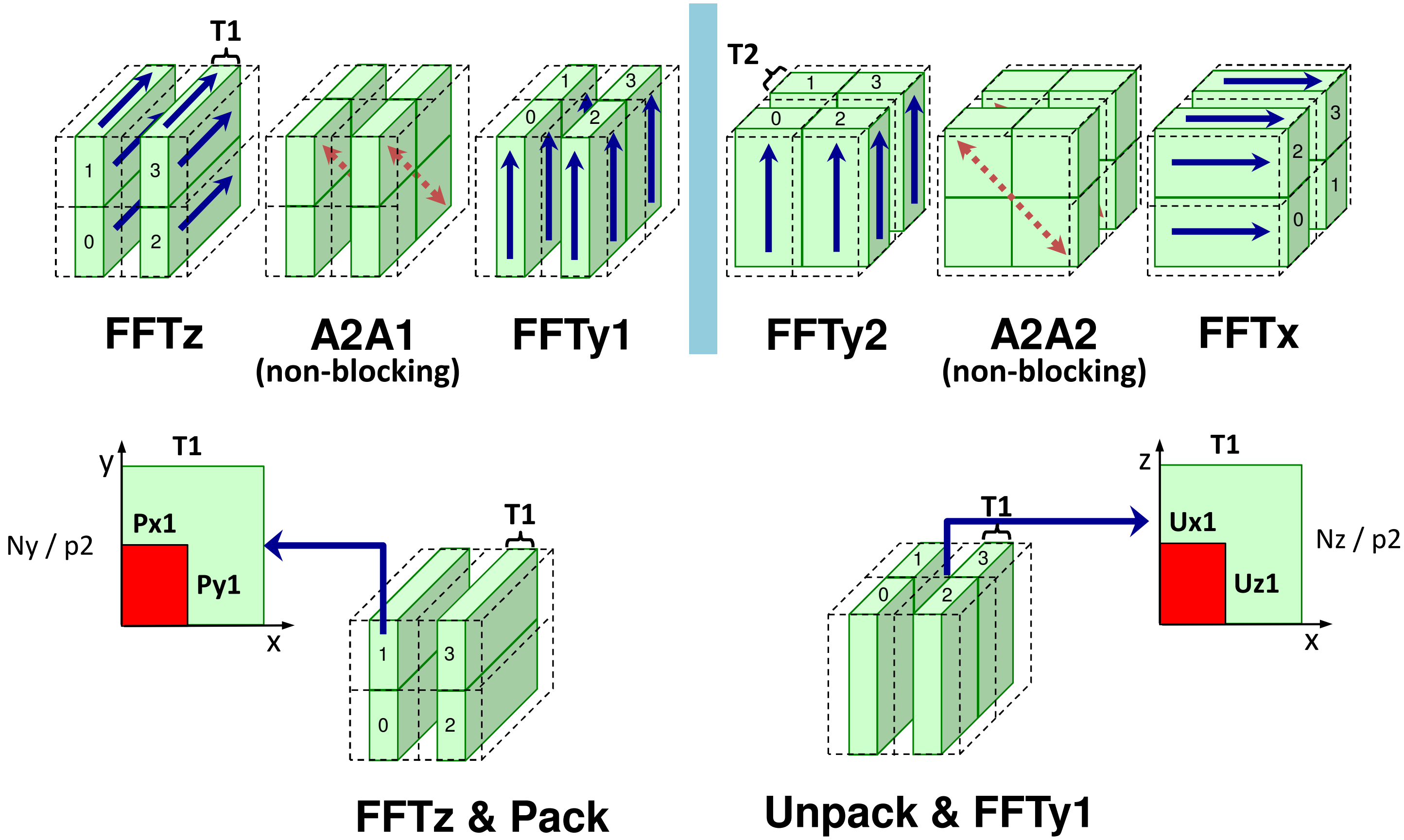
- FFT in 3 dimensions
  – Composed of three 1 dimensional FFT's
  – Data is redistributed among processes between FFT's



Computation/Communication Overlap



Auto-tuning Opportunities



Online Auto-Tuning

Active Harmony Wel

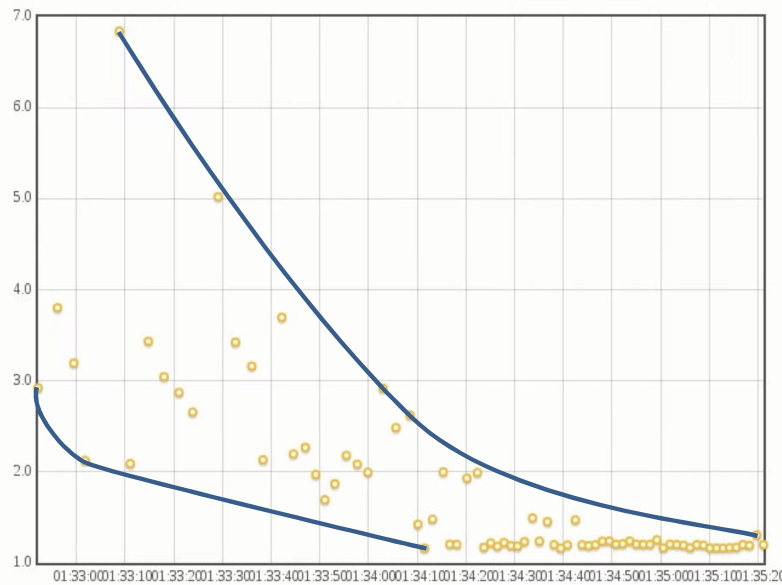
bryce.cs.umd.edu:1979/session-view.cgi?persist

Session Name:	persist
Session Strategy:	nm.so
Session Status:	Active - Searching
Connected Clients:	1

Restart Point: Restart Pause Resume Kill

Refresh Interval: 1 4/6/2016 4:35:22 AM
View: Timeline Chart Size: 640x480
Table Length: 10 Report Precision: 4

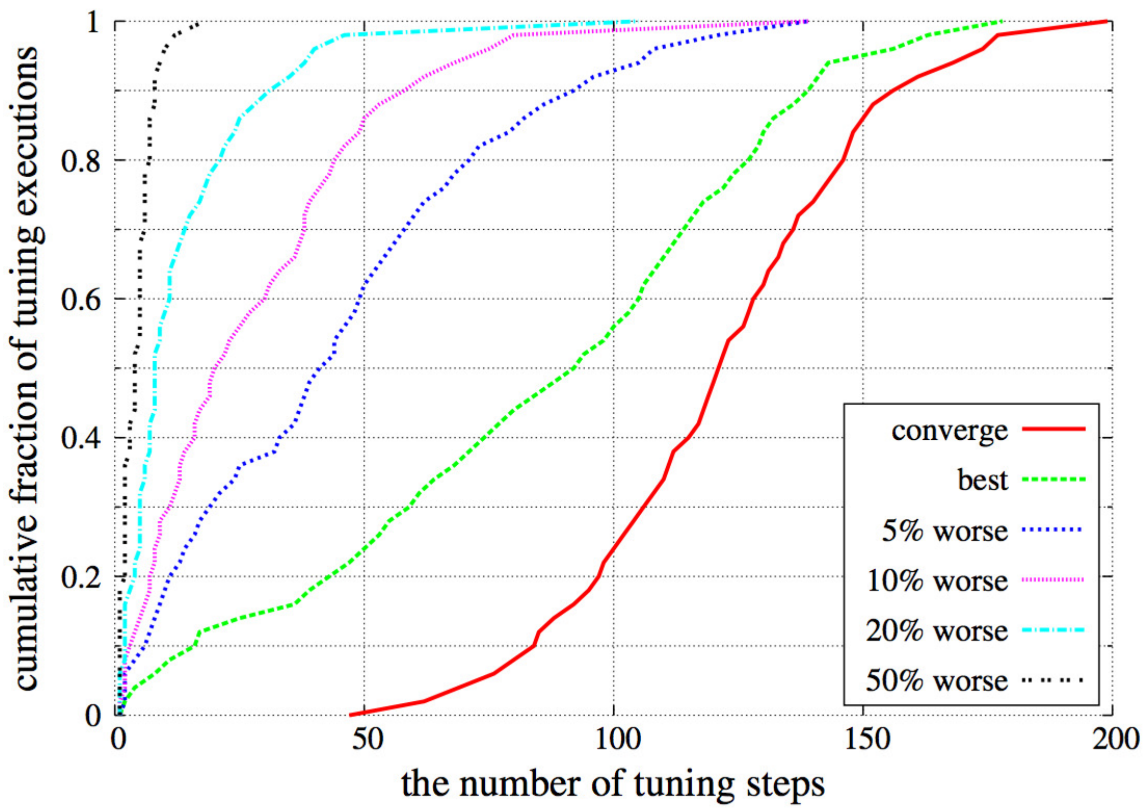
	P1	Ry	W1	W2	T1	T2	V	S	Fz	FP1	FU1	FU2	Fy1	Fy2	FP2	Fx	Performance
Best	2	6	7	3	9	8	1	0	7	5	11	10	9	9	4	9	1.162
4:35:21 AM	2	6	6	4	8	8	1	0	8	7	10	10	9	9	5	10	1.200
4:35:19 AM	2	5	6	3	9	8	1	0	7	6	9	11	9	9	4	9	1.307
4:35:18 AM	3	6	6	4	8	7	1	0	8	6	10	9	10	10	5	10	1.193
4:35:16 AM	2	5	6	4	8	8	1	0	9	5	10	12	10	9	5	9	1.200
4:35:15 AM	2	5	6	4	9	8	1	0	6	6	10	11	10	8	5	9	1.170
4:35:14 AM	2	6	6	4	9	8	1	0	8	6	10	11	8	8	5	10	1.167
4:35:12 AM	2	6	7	4	9	8	1	0	8	6	8	11	9	8	5	10	1.163
4:35:11 AM	2	5	7	4	8	7	1	0	8	7	10	11	9	9	5	9	1.164
4:35:10 AM	2	5	6	3	8	7	1	0	8	7	11	10	10	8	6	11	1.164
4:35:08 AM	3	5	6	4	8	8	1	0	8	6	10	11	8	9	6	10	1.195



Phase Aware Tuning

- Improvements over offline (non-phase) tuning
  - Reduce search dimensions from 24 to 16
  - 40% fewer search steps needed to converge
  - Equivalent performance after convergence
- Eliminates need for training runs
  - Don't allocate thousands of nodes to train

Offline Auto-Tuning Cost



Conclusion

- Phases are key for HPC analysis tools
  - Rely on human guidance through annotations
  - Virtualizing repeated phases helps many types of tools
- Annotations unite cross-domain expertise
  - Libraries annotate variables to analyze
  - Application annotate regions to analyze
- Currently analyzing other HPC codes