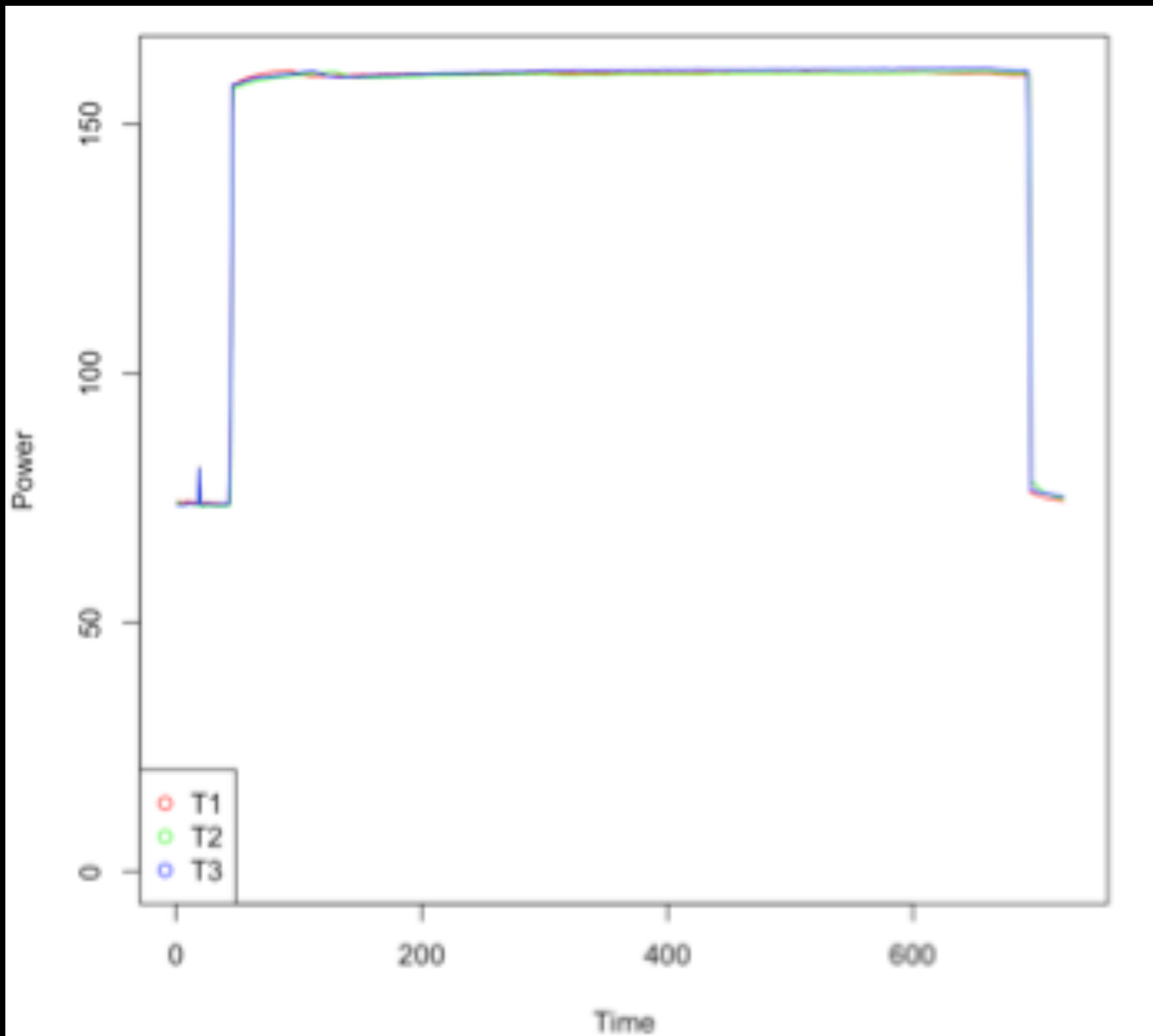
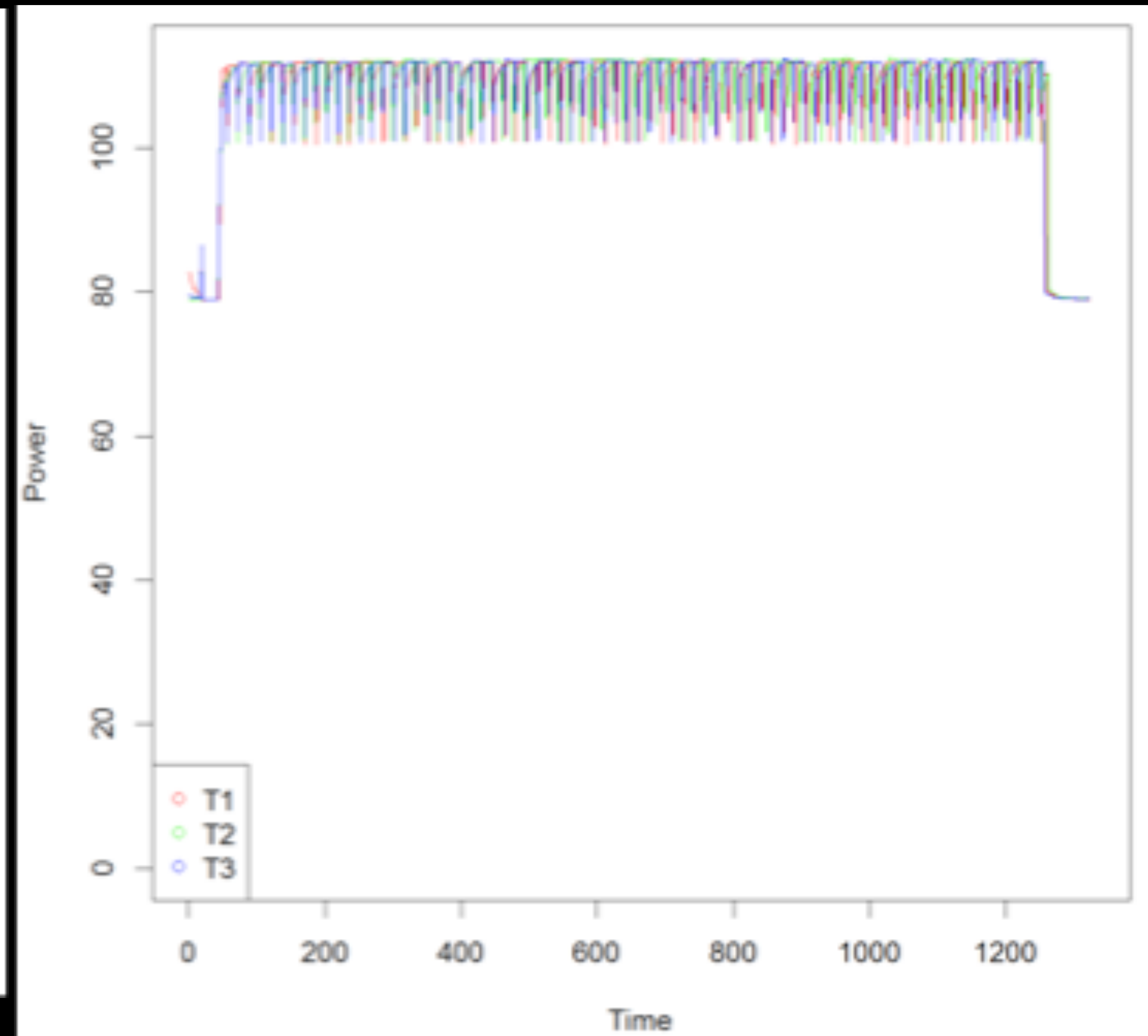
Power Signatures of HPC Workloads

Suzanne Rivoire, Sonoma State University

Second Workshop on HPC Power Management: Knowledge Discovery

August 25, 2016


Power
Time
0
50
100
150
0
200
400
600
T1
T2
T3

Power
Time
0
20
40
60
80
100
0
200
400
600
800
1000
1200
T1
T2
T3

Motivation

Prior Work in HPC

(all from visual inspection of power traces)

- **Song et al., 2009:** Application-specific patterns from run to run and (less so) across machines
- **Laros et al., 2009:** 2 traces from same application on different platforms look alike
- **Kamil et al., 2008:** Can possibly distinguish *classes* of workloads
  - but CPU-intensive workloads all look like LINPACK

Our Questions

- Do applications exhibit distinctive power consumption behavior, even across different
  - runs
  - input data
  - hardware platforms or resources?
- Can we identify an application from its power trace?
- ...and can we do this **automatically**?

Who cares?

- Patterns => application-specific power optimizations
- Recognizing a job from power traces => make better resource allocation decisions

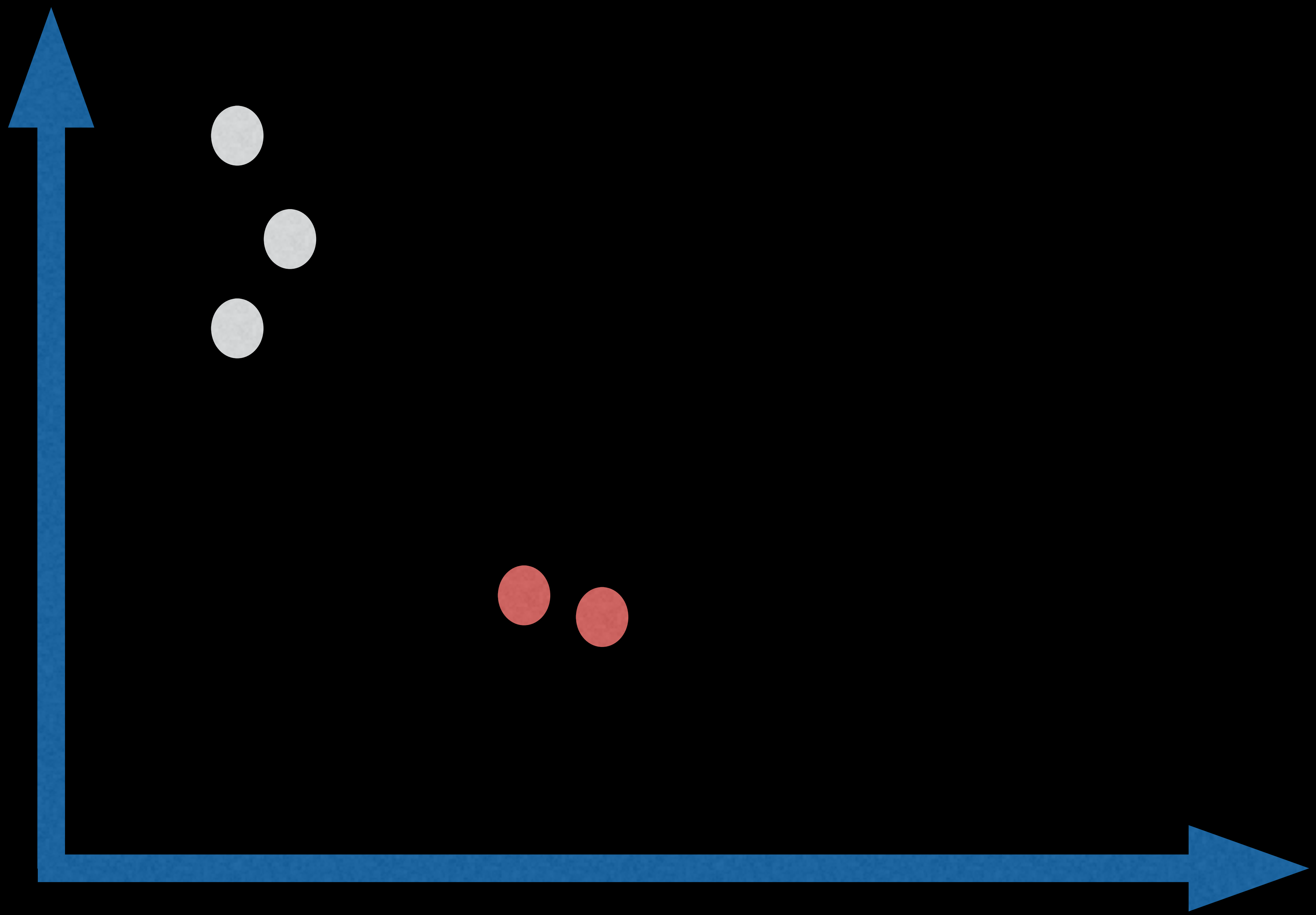
Our Context

- Examples of prior work
  - Identifying websites visited by a Mac Mini from power traces sampled at 250 KHz [Clark et al.]
  - Periodicity in long-term power traces from cloud providers [Wang et al., Herbst et al.]
- HPC is different!
  - Time granularity of measurements
  - Probably less periodic in general; definitely masked at 1 Hz

Outline

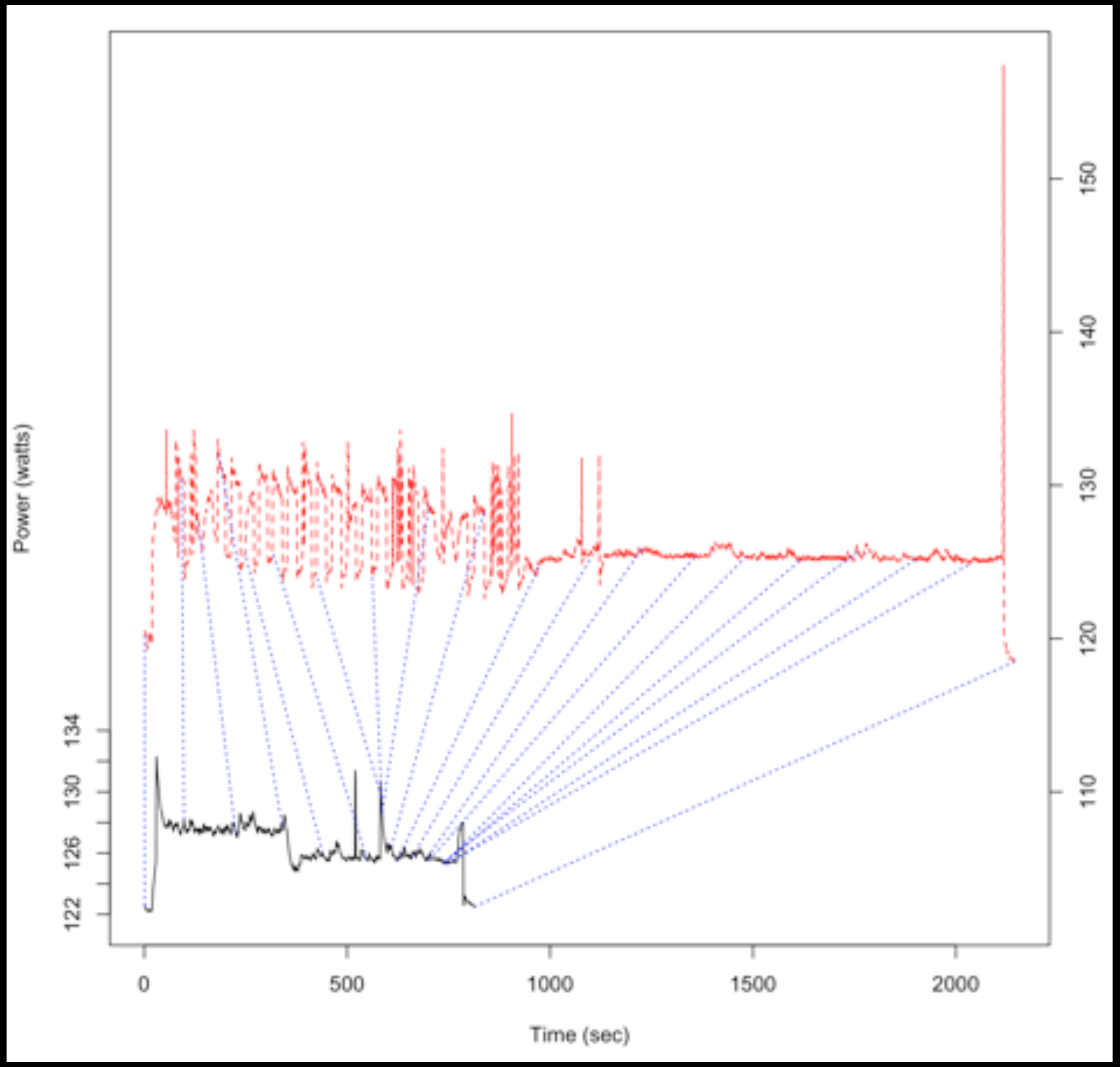
- **Clustering:** Are there patterns?
- **Classification:** Recognizing an application from its power trace
- **Novelty detection:** Using "none of the above" to identify new workloads
- Current & future work

Clustering

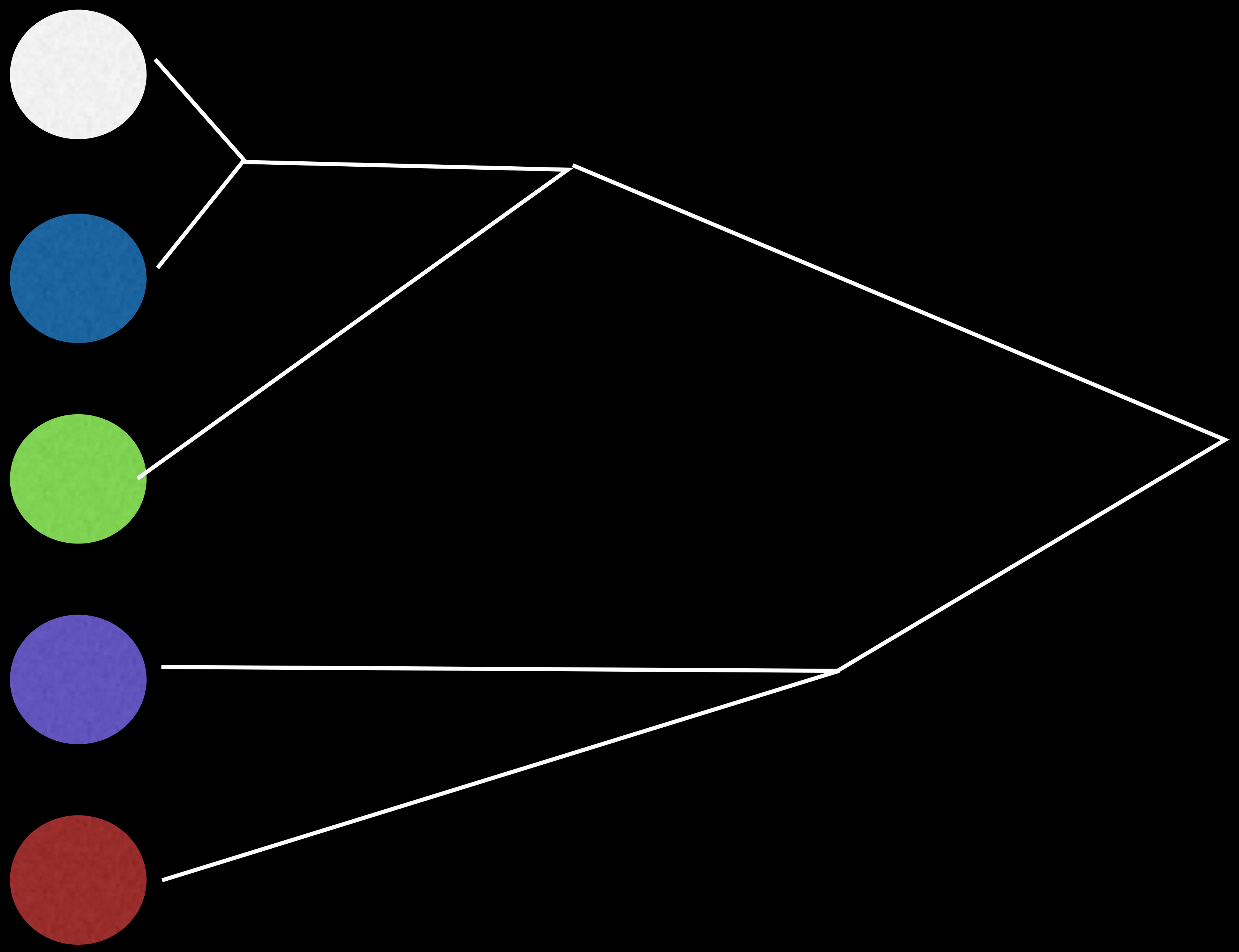


Representations and Distance Metrics

- Time series
  - Mean squared distance
  - Dynamic time warping
- Feature-based



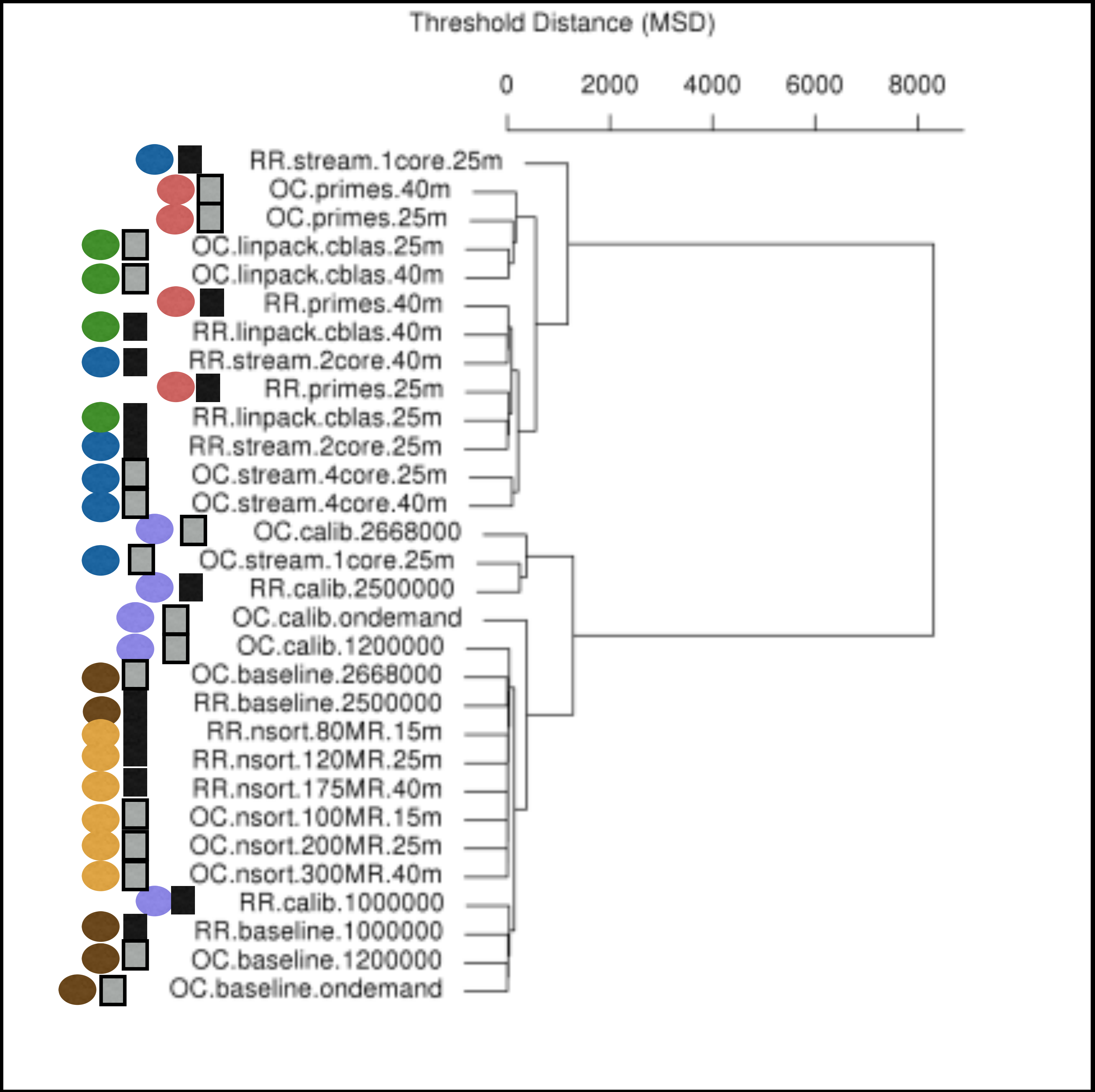
Hierarchical Clustering



hclust Results

Circle color: workload
Square color: machine

Very clean 2-clustering (CPU-intensive vs. not)

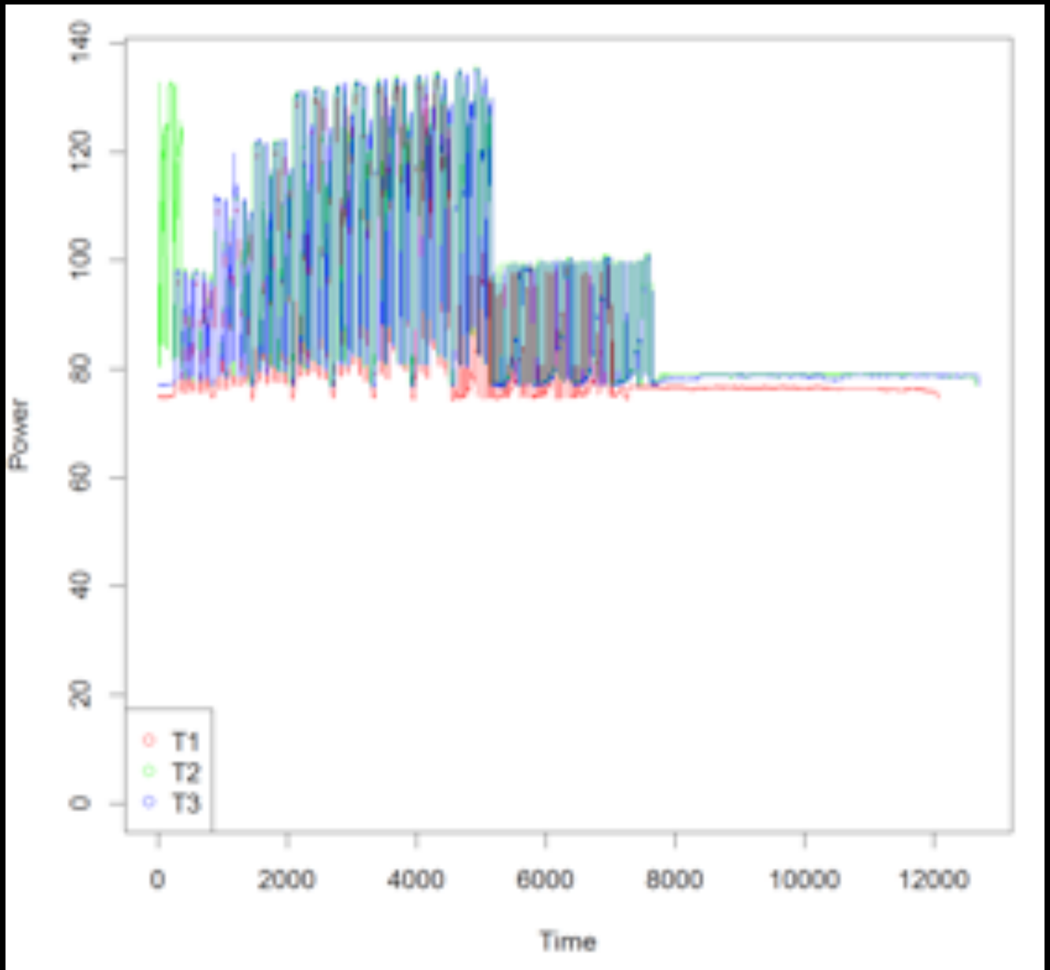


Quantitative Validation

- Quantifying clustering goodness is surprisingly complicated -- see Combs et al., E2SC 2014
- Takeaways from a larger (220-trace) dataset focused on CPU-intensive kernels:
  - There is a signal here
  - Feature vectors work as well as DTW and are much cheaper in space and **time**

Feature vectors

Input: set of power traces labeled by workload



Output: set of signatures, one per trace, plus workload label

[DCSkewness, DCKurtosis, DCNonlinearity, DCSerialCorrelation, hurst, kurtosis, lyapunov, max, mean, median, min, nonlinearity, skewness, standard_deviation, serial_correlation, trend, (workload)]

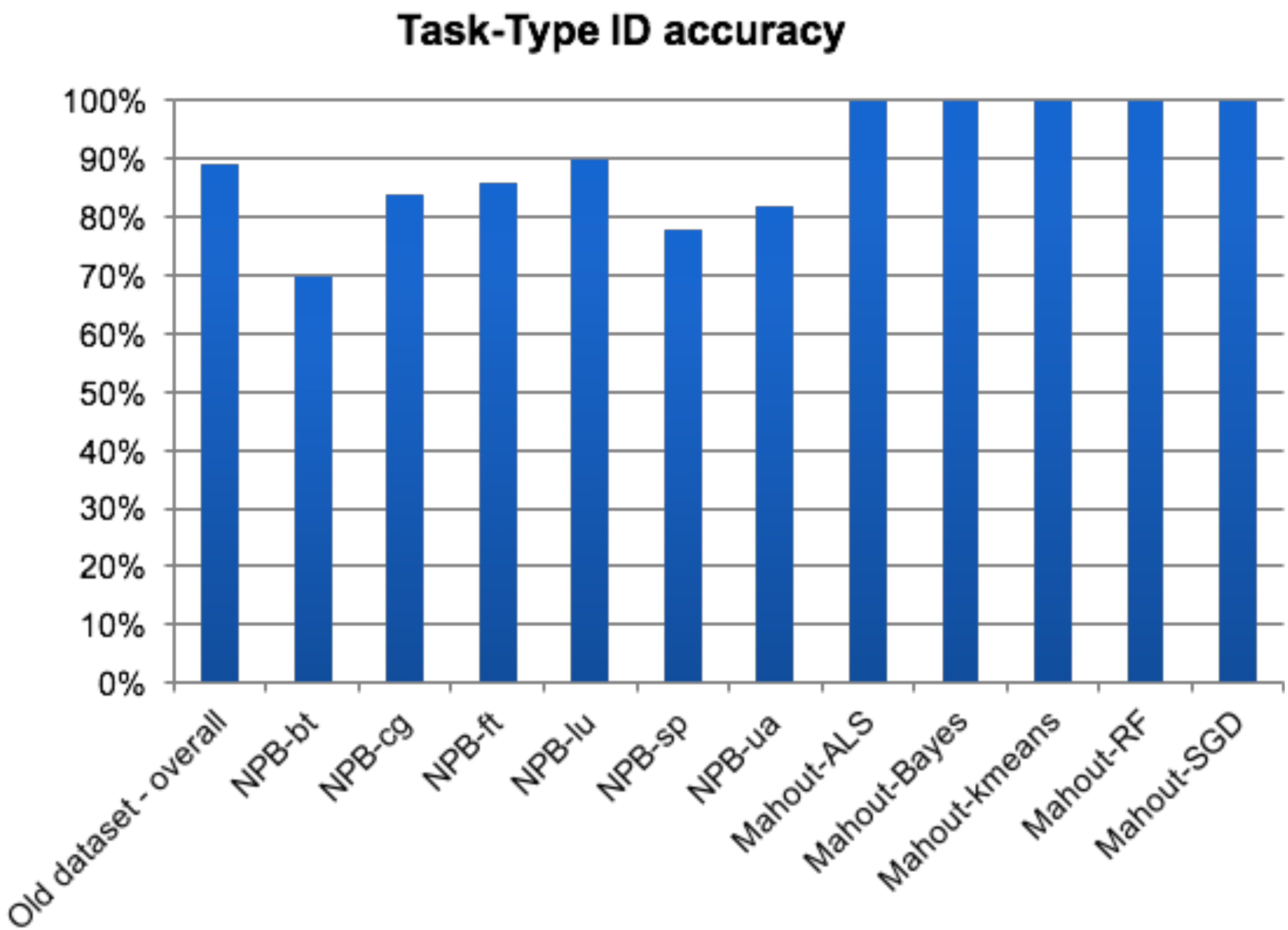
from Wang et al, 2006

Classification

- Given a set of traces from known workloads, can we identify the workload of an unlabeled trace?
- Approach: random forest [Breiman '01]
    - Automatically build a bunch of decision trees and let them vote
- ~90% accurate for original 220-trace set

Additional Workloads

- NPB workloads: serial, MPI, OMP with different #s threads
- Mahout big data analysis workloads


Task-Type ID accuracy
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Old dataset - overall
NPB-bt
NPB-cg
NPB-ft
NPB-lu
NPB-sp
NPB-ua
Mahout-ALS
Mahout-Bayes
Mahout-kmeans
Mahout-RF
Mahout-SGD

Novelty Detection

- Given an unknown power trace, identify its workload ***or say "None of the above"***
- Helpful for identifying new / emerging task types

Approach: Metaclassifier

- That forest of decision trees is full of information -- let's use it!
- Input to novelty detector is *predictions* from workload classifier: how is the forest different for known vs. unknown workloads?

Example: Certainty

Average Results

- **Precision:** when we call something a novelty (or a known), are we right?
- **Recall:** are we finding all the novelties (or knowns) in the dataset?

	Known	Novelty
Precision	72.4%	80.4%
Recall	83.3%	68.4%

Current and Future Work

- **Phase detection:** can we identify phases of a power trace?
- **Early classification:** can we identify a power trace online, while the workload is still running?
- **Dataset evaluation:** how to quantify trace complexity or dataset completeness?

Conclusion

- Applications exhibit distinctive power consumption behavior, even across datasets and machines
- Compact feature vectors are enough to ID an application from its power trace
- …or to identify an unknown application

Wish List

- **Two words: Ubiquitous instrumentation**
- Power sensors, with as high a sampling rate as possible
    - Per node
    - Per job
- Synchronization of power data...
    - With workload start and end times
    - With internal counters like RAPL
- The dream: having this instrumentation on low-end/mobile systems too, since power optimizations in these domains filters into HPC

Collaborators

Chung-Hsing Hsu

Nick Armour	Brett Granborg	Lowell Olson
Mark Ayala	Matt Hardwick	Jacob Probst
Jeff Bahns	Gaybi Igno	Stephen W. Poole
Jorge Cabrera	Alex Katz	Kelsey Rangel
Shin Mei Chan	Ryan Kelez	Fabian Santiago
Aaron Cheever	Joji Kubota	David Tran
Jacob Combs	Sean McKinley	Rachelle Thysell
Joe Granados	Jolie Nazor	Sarah Whitaker

This work was supported by the United States Department of Defense (DoD) and used resources of the Computational Research and Development Programs at Oak Ridge National Laboratory.