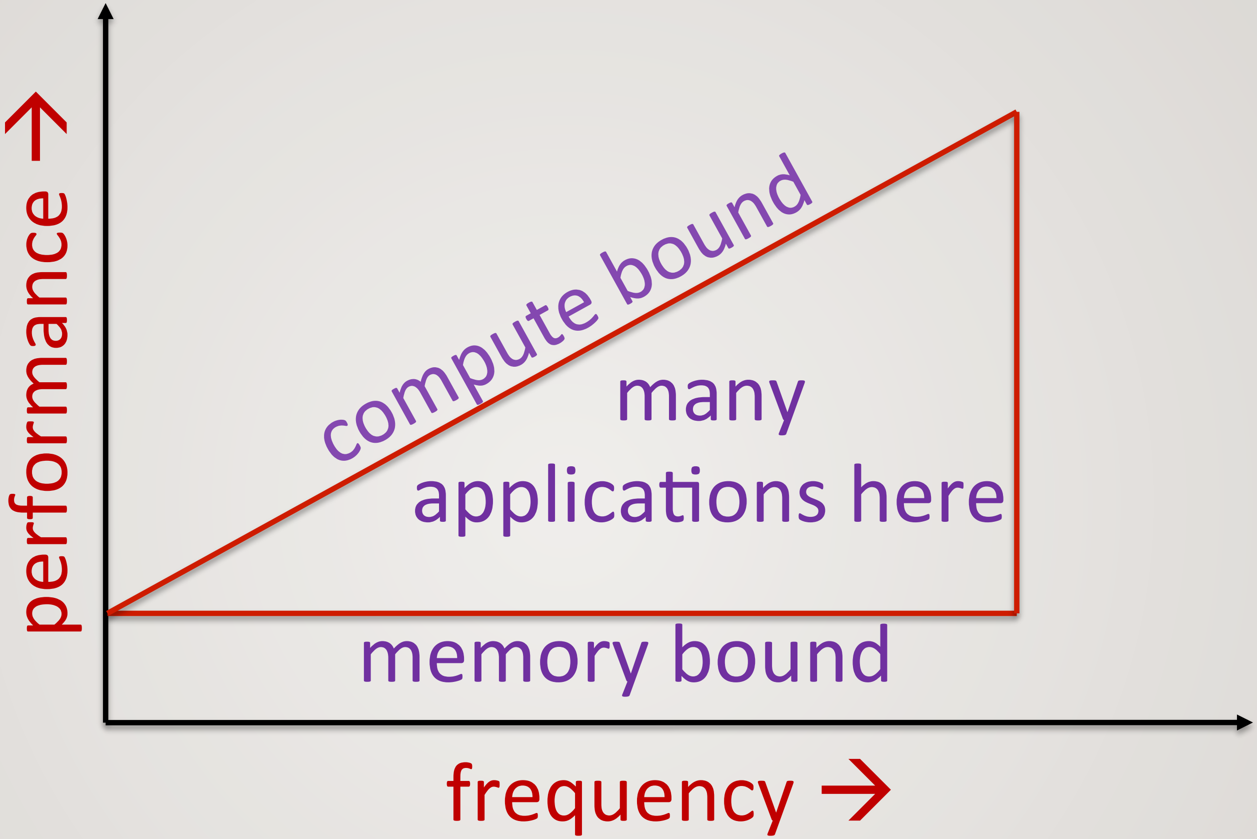
DVFS PERFORMANCE PREDICTION FOR MANAGED MULTITHREADED APPLICATIONS

Shoaib Akram, Jennifer B. Sartor, Lieven Eeckhout
Ghent University, Belgium
Shoaib.Akram@elis.UGent.be

DVFS Performance Prediction



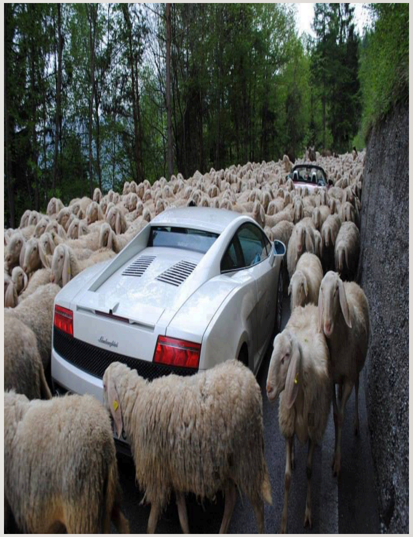
Sample at all DVFS states ☹

Estimate performance ☺

Managed Multithreaded Applications

Heterogeneity

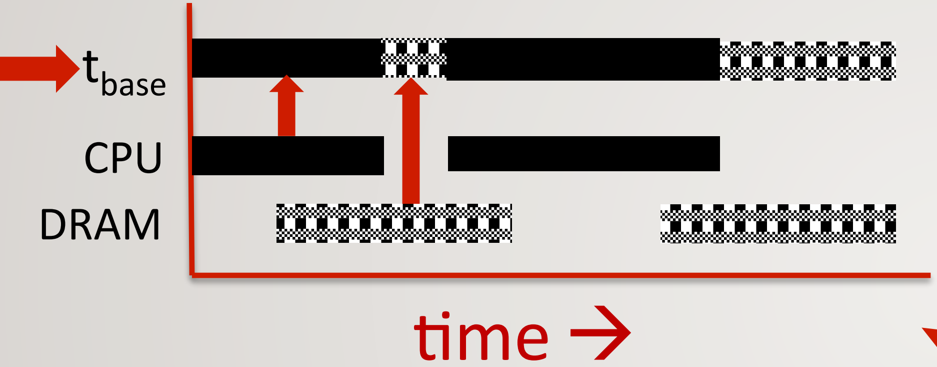
Synchronization



Store Bursts

Background

Base Frequency



- t_{base} sum of
  - Scaling (S)
  - Non-Scaling (NS)

Target Frequency

- r = Base/Target
- S → S * r
- NS → No change
- $t_{target} = (S*r) + NS$

- Not simple
- OOO+MLP

State of the Art

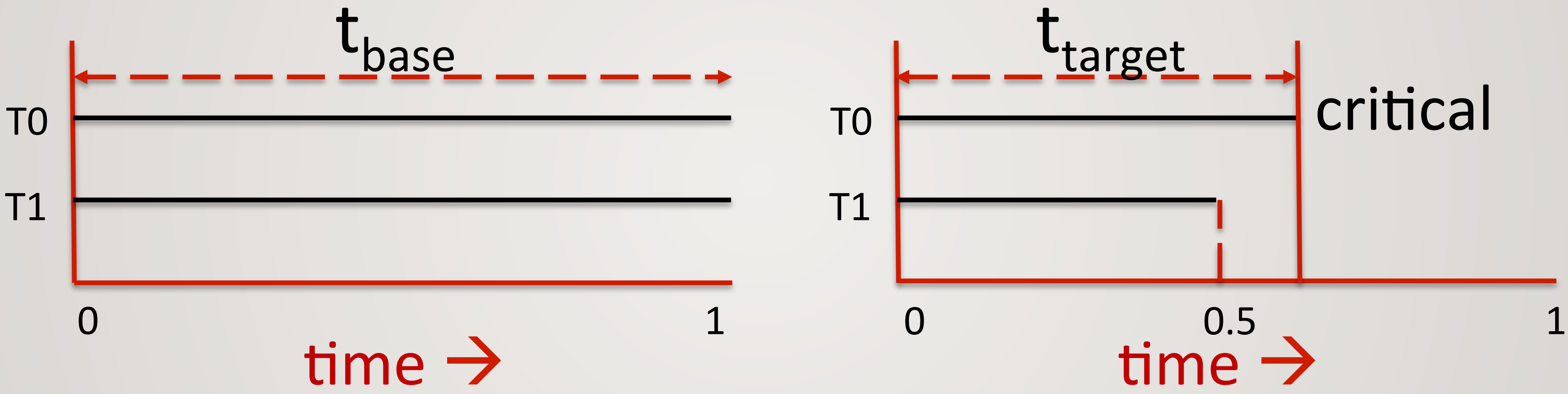
- **CRIT** estimates non-scaling by
  - Measuring critical path through loads
  - Ignoring store operations

R. Miftakhutdinov, E. Ebrahimi, and Y. N. Patt. Predicting performance impact of DVFS for realistic memory systems. MICRO, 2012.

Multithreaded CRIT (M+CRIT)

Base Frequency → Target Frequency

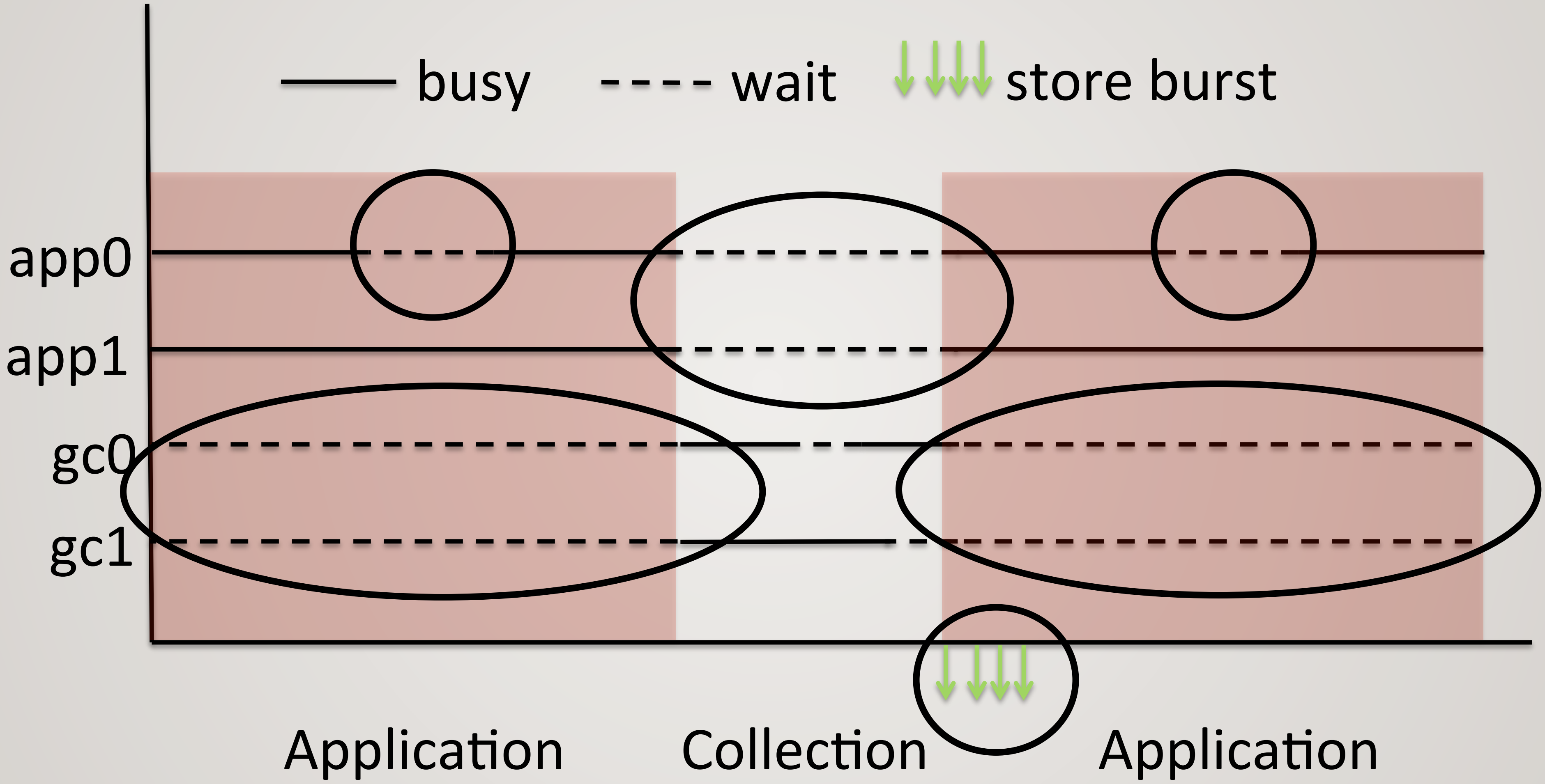
2X



Use CRIT to identify each thread's non-scaling

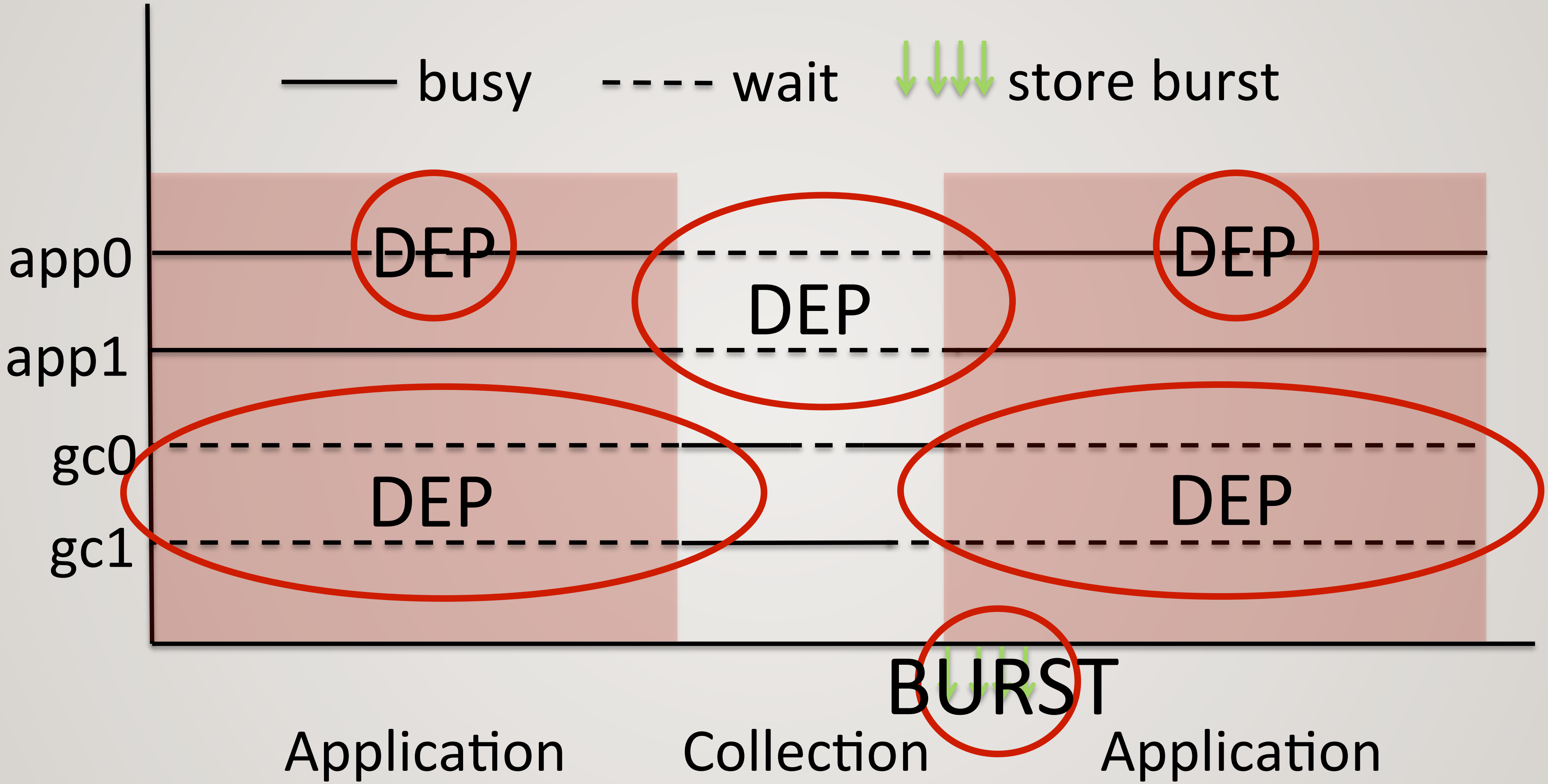
High error for multithreaded Java!

Sources of Inaccuracy in M+CRIT



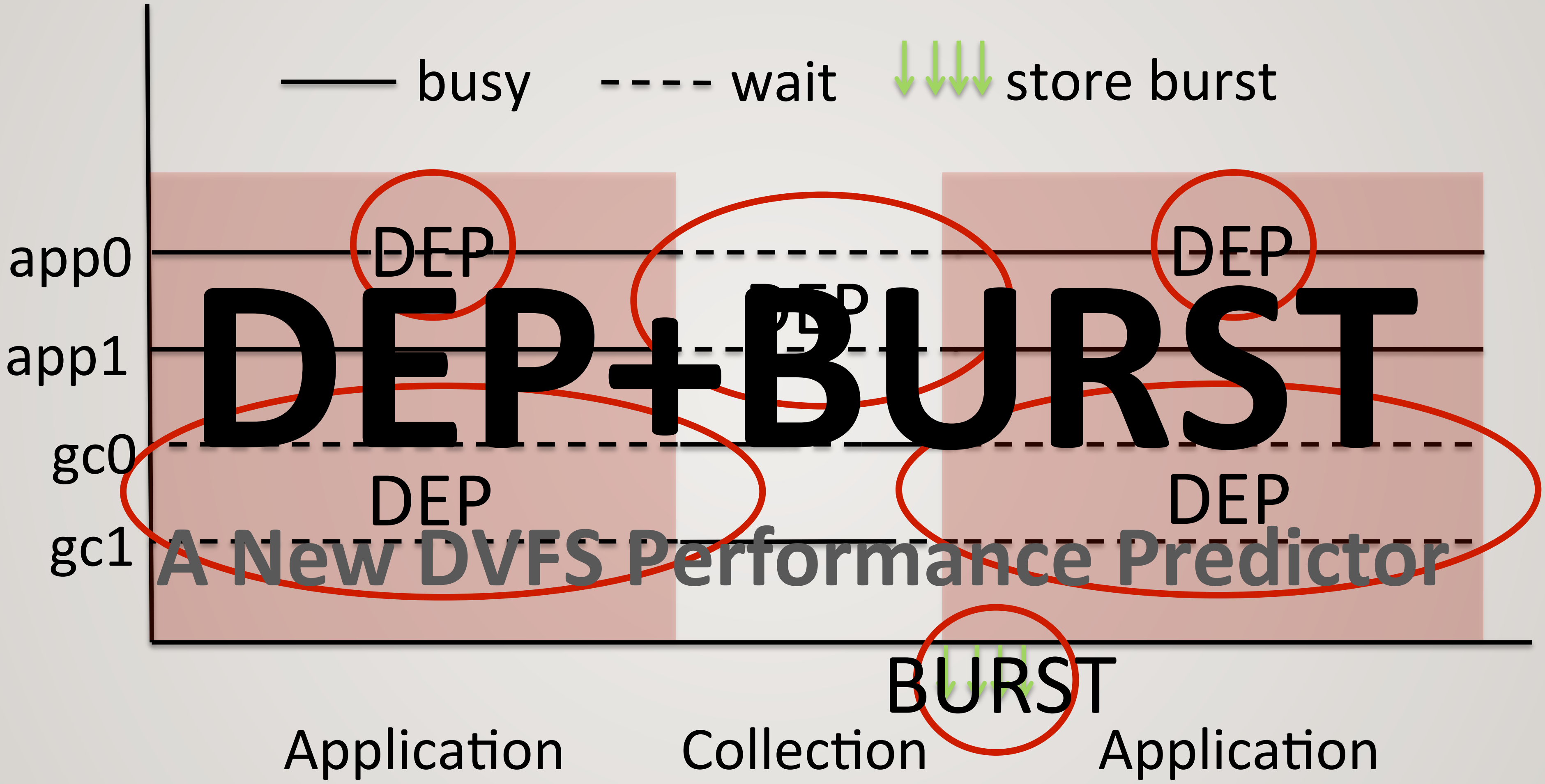
Scaling or non-scaling?

Sources of Inaccuracy in M+CRIT



Scaling or non-scaling?

Our Contribution



DEP+BURST

A New DVFS Performance Predictor

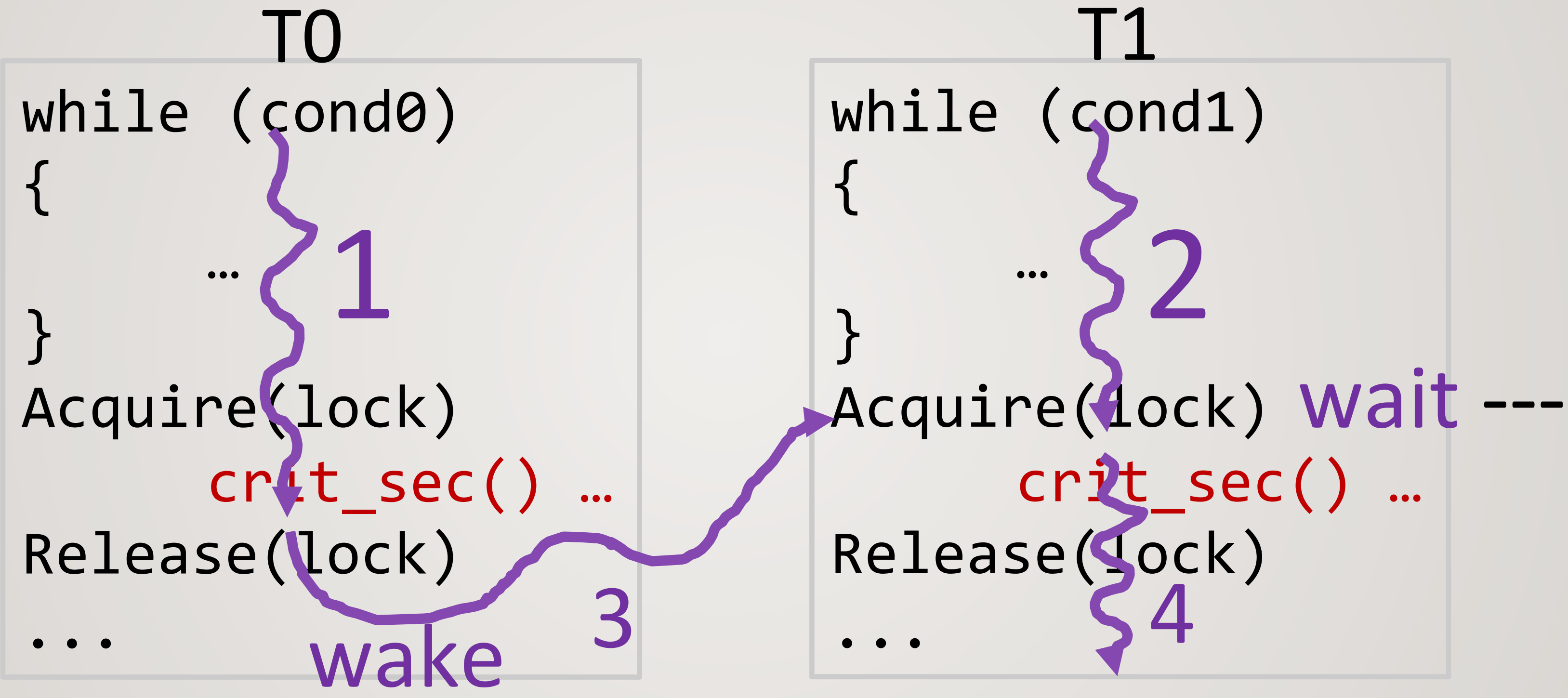
Scaling or non-scaling?

Our Contribution

DEP+BURST

A New DVFS Performance Predictor

Example: Inter-thread Dependences



T0

```
while (cond0)
{
        …
}
Acquire(lock)
        crit_sec() …
Release(lock)
...
```

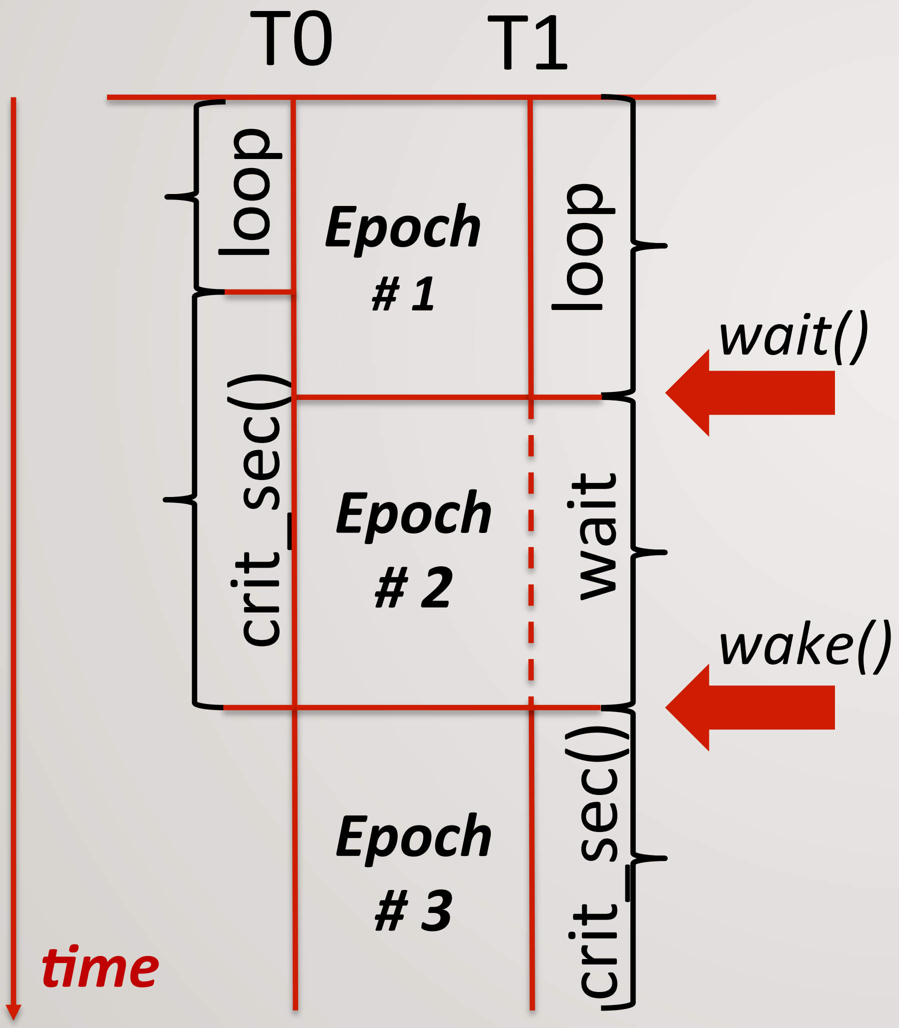
T1

```
while (cond1)
{
        …
}
Acquire(lock)
        crit_sec() …
Release(lock)
...
```

- Intercept synchronization activity
- Reconstruct execution at target frequency

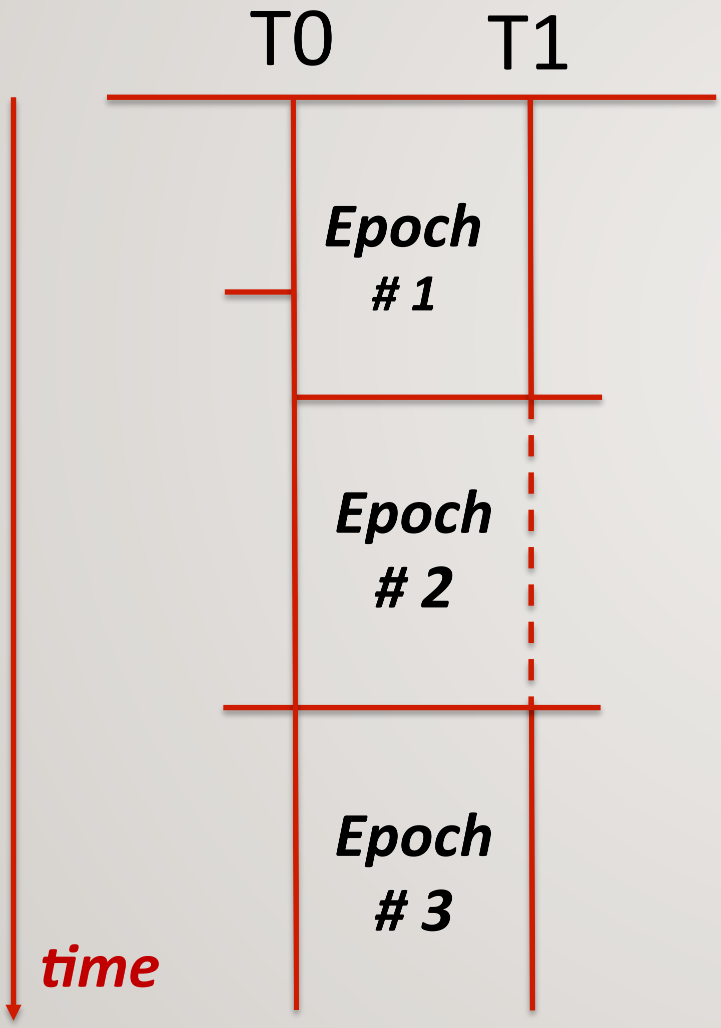
Identifying Synchronization Epochs

Base Frequency → Target Frequency



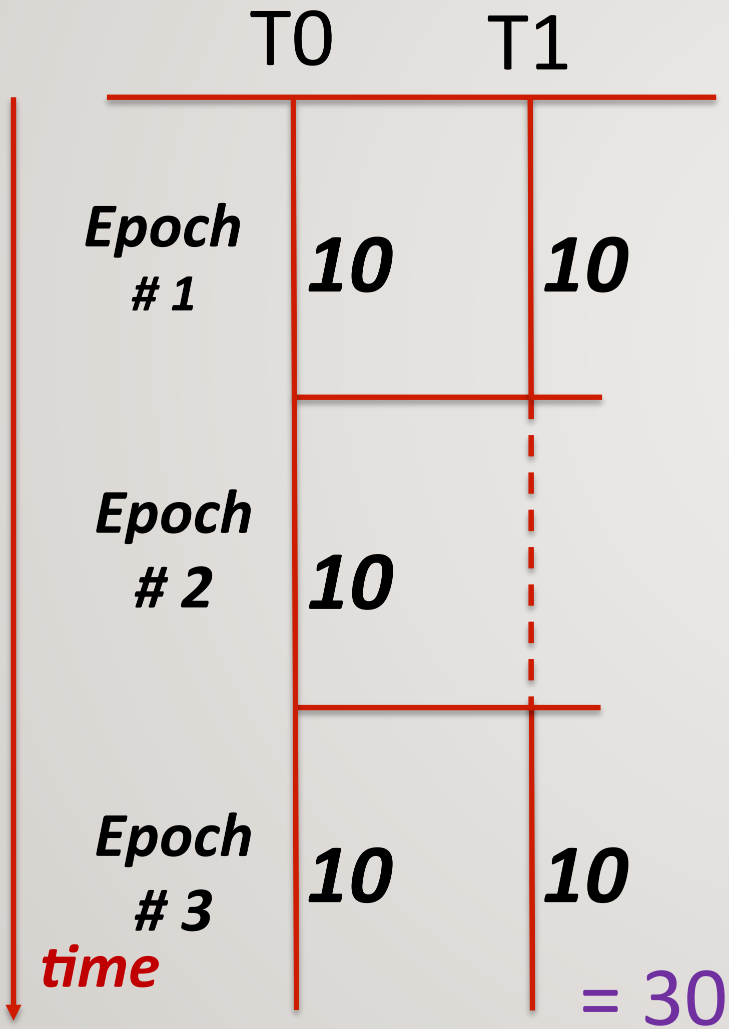
Identifying Synchronization Epochs

Base Frequency → Target Frequency



Identifying Synchronization Epochs

Base Frequency → Target Frequency



T0

T1

Epoch # 1

10

10

Epoch # 2

10

Epoch # 3

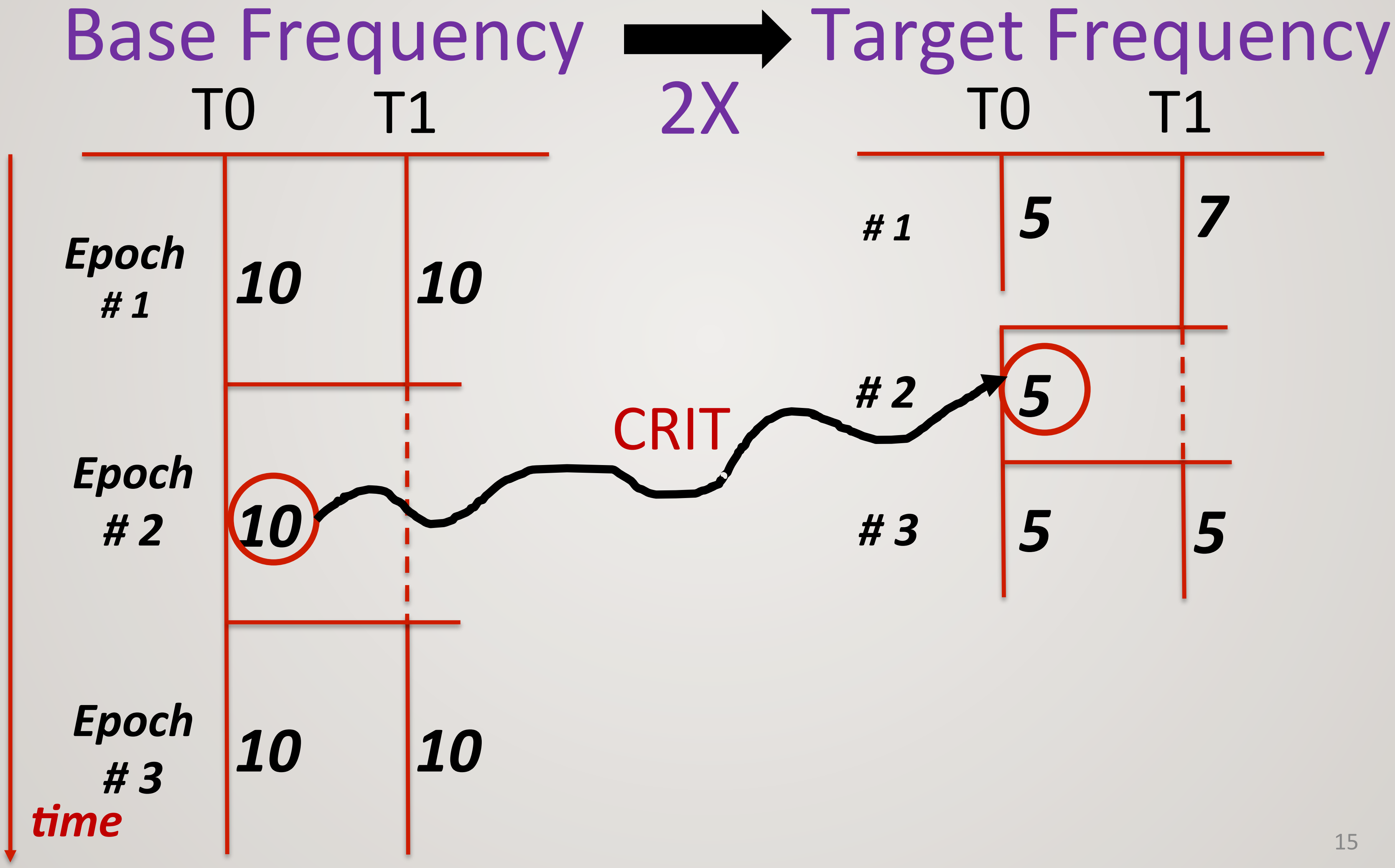
10

10

time

= 30 units

Reconstruction at Target Frequency



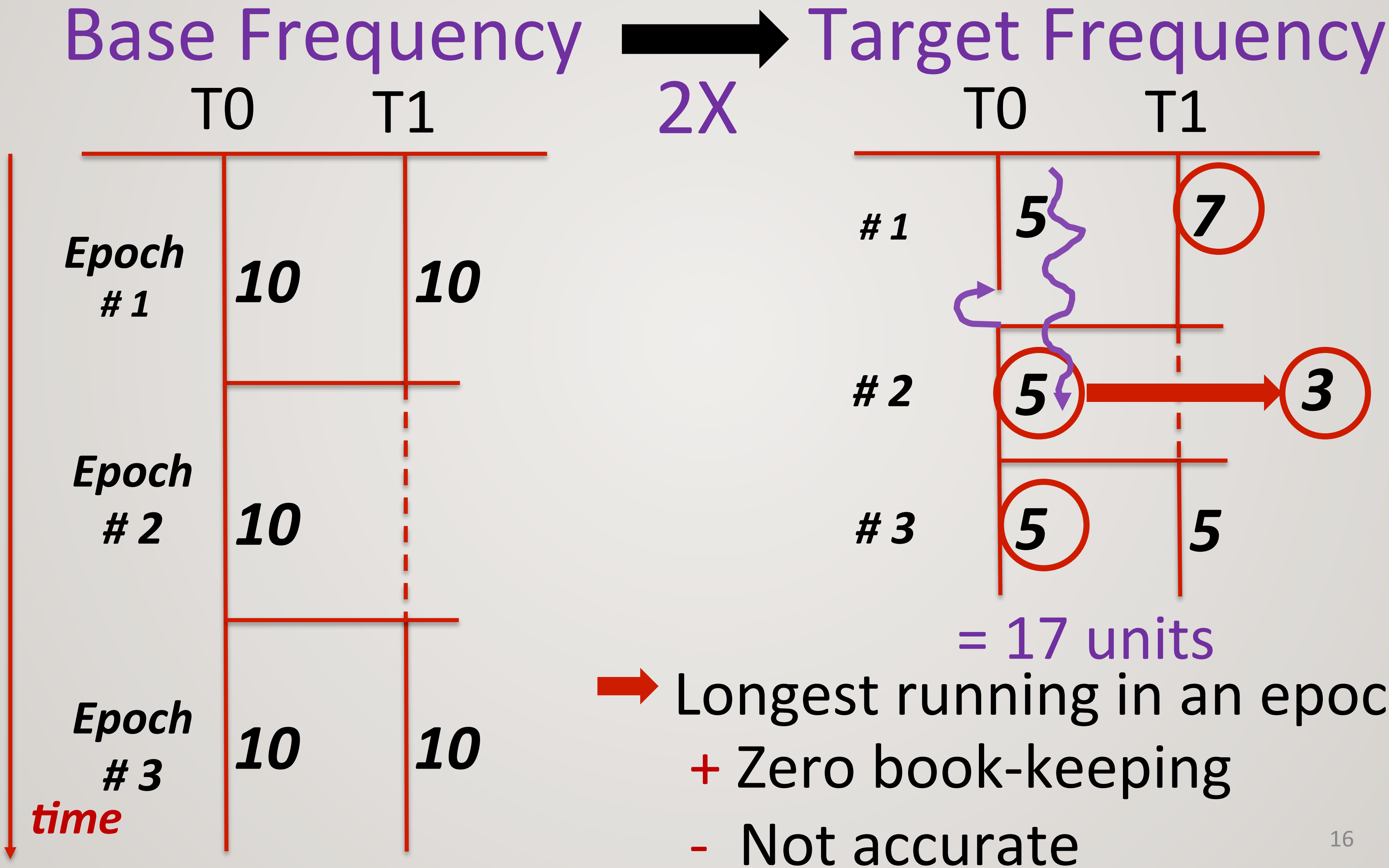
Base Frequency → Target Frequency

2X

	T0	T1
Epoch # 1	***10***	***10***
Epoch # 2	***10***	
Epoch # 3	***10***	***10***

	T0	T1
# 1	***5***	***7***
# 2	***5***	
# 3	***5***	***5***

Reconstruction at Target Frequency

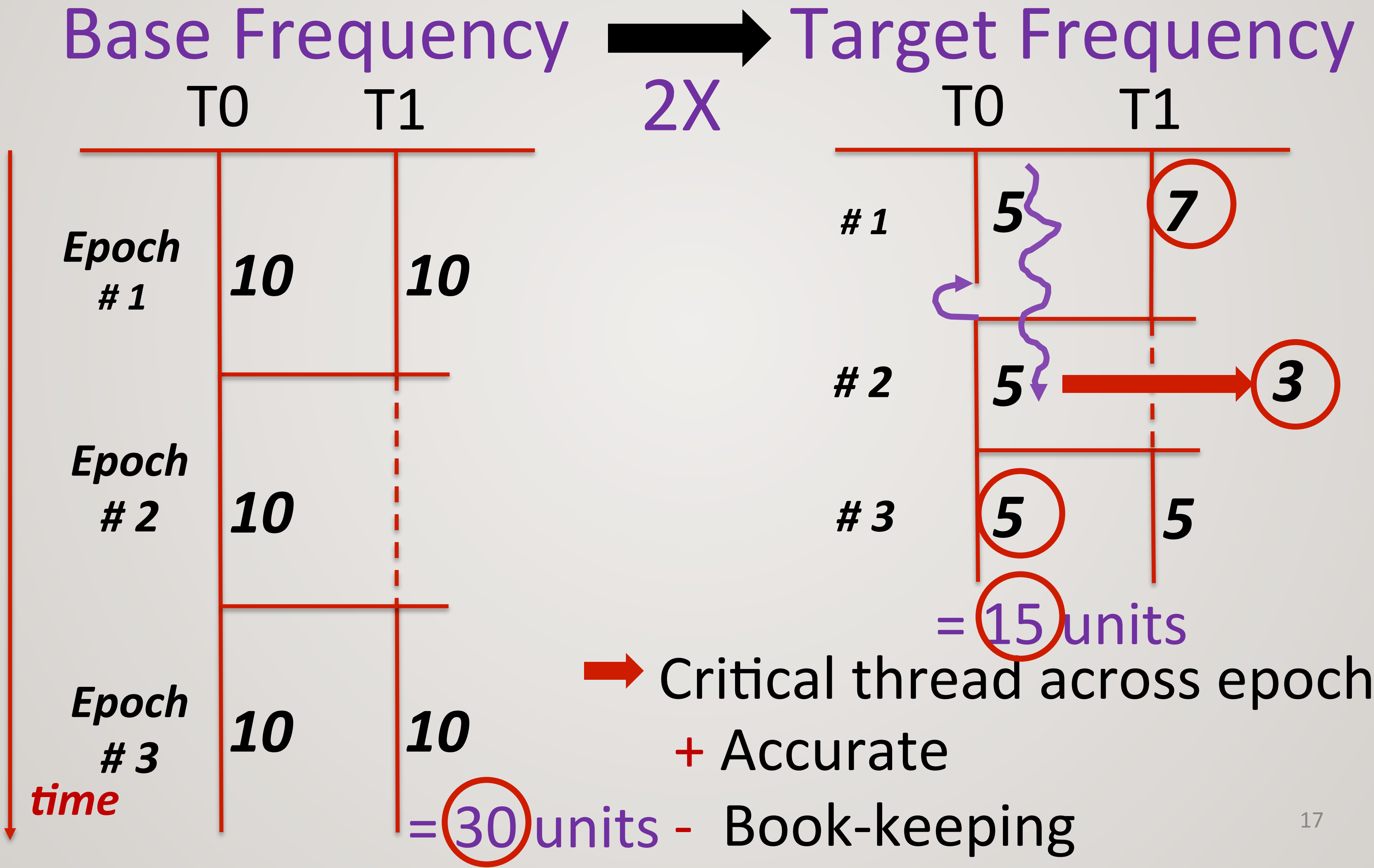


➡ Longest running in an epoch

+ Zero book-keeping

- Not accurate

Reconstruction at Target Frequency



Base Frequency → Target Frequency

2X

T0 T1

Epoch # 1: 10 10

Epoch # 2: 10

Epoch # 3: 10 10

time

= 30 units

T0 T1

1: 5 7

2: 5 3

3: 5 5

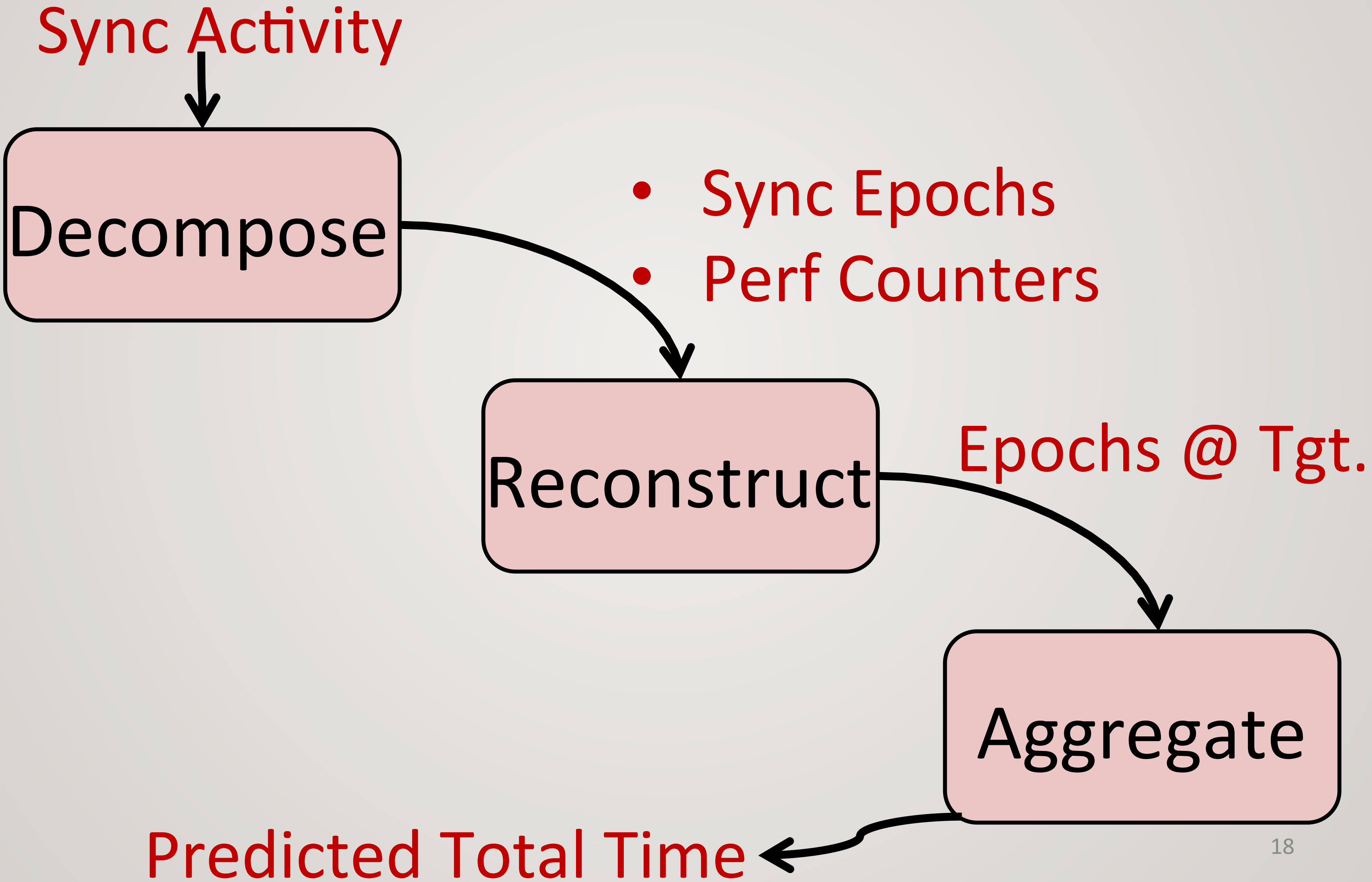
= 15 units

➡ Critical thread across epochs

+ Accurate

- Book-keeping

DEP: Summary



Sync Activity

Decompose

- Sync Epochs
- Perf Counters

Reconstruct

Epochs @ Tgt.

Aggregate

Predicted Total Time

Our Contribution

DEP+BURST

A New DVFS Performance Predictor

Our Contribution

DEP+BURST

A New DVFS Performance Predictor

Store Bursts

- Reasons
  - Zero initialization
  - Copying collectors
- Modeling Steps
  - Track how long the store queue is full
  - Add to the non-scaling component

Methodology

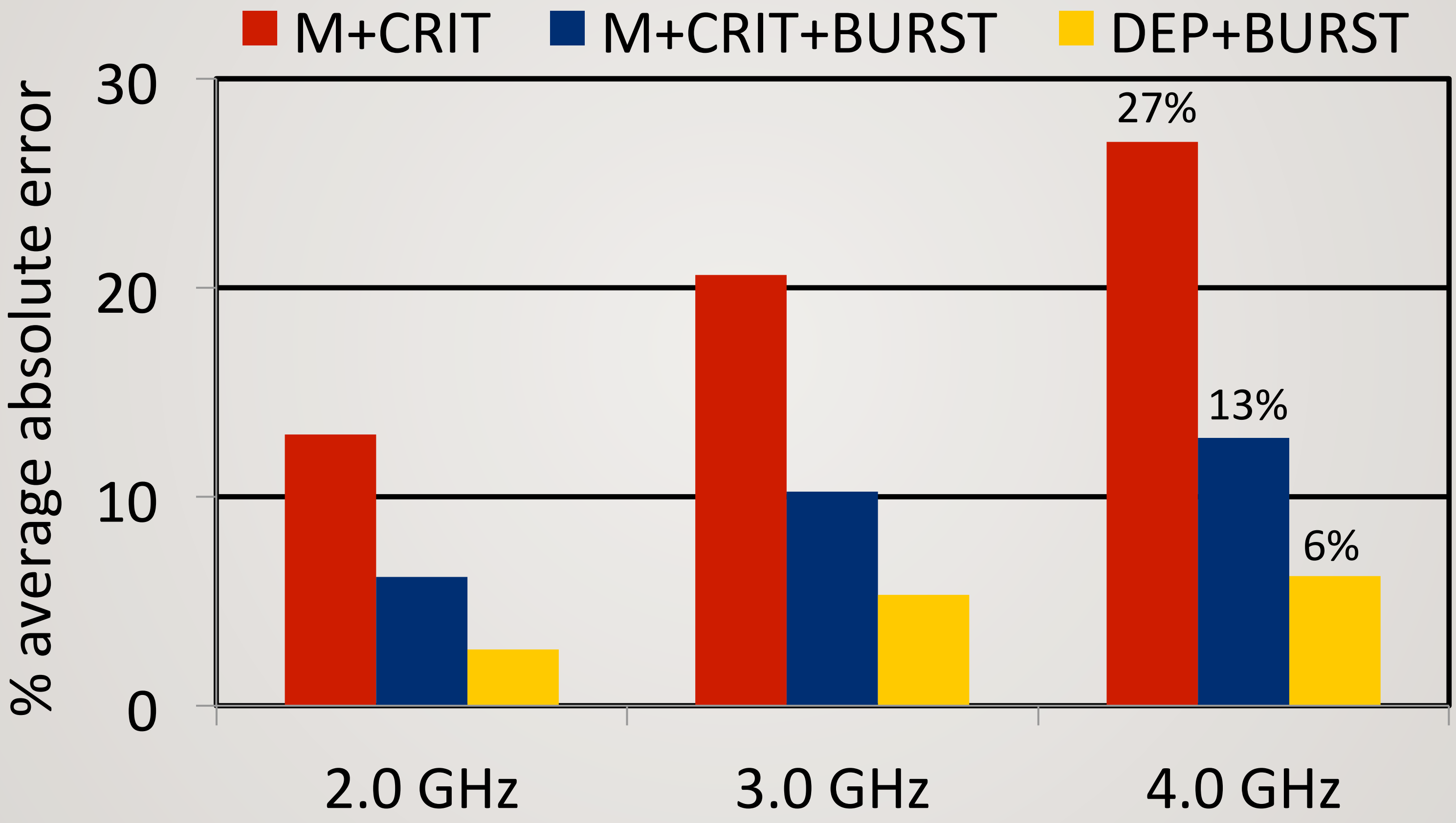
- Jikes RVM 3.1.2
- Production collector (Immix)
- # GC threads = 2
- 2x min. heap

Version 6.0

- 4 cores, 1.0 GHz → 4.0 GHz
- 3-level cache hierarchy
- LLC fixed to 1.5 GHz
- DVFS settings for 22 nm Haswell

- Seven multithreaded benchmarks
- Four application threads

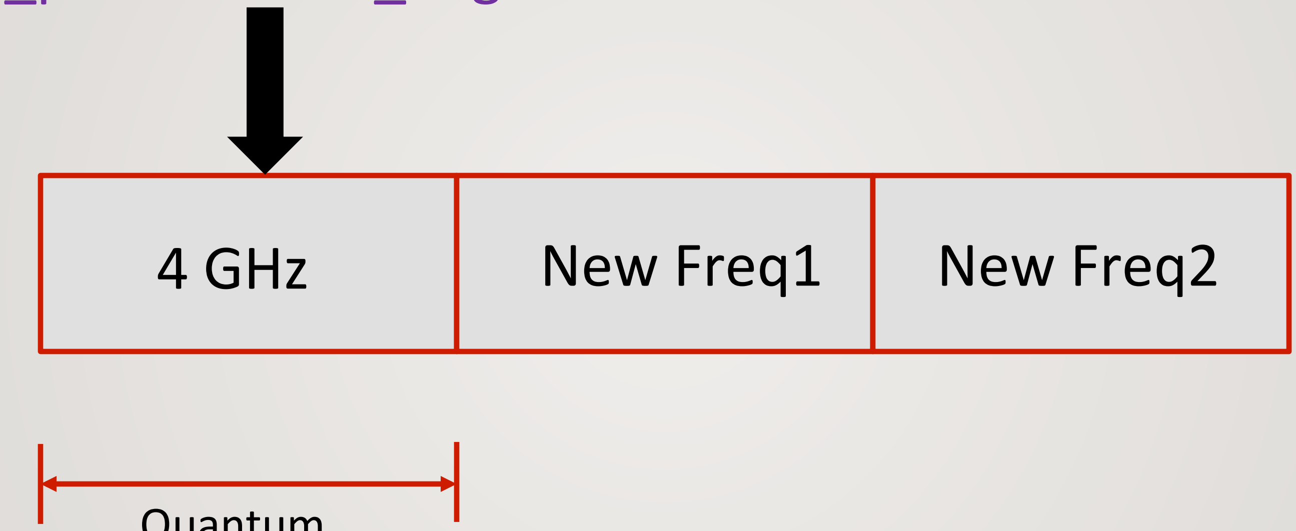
Accuracy



Baseline Frequency = 1.0 GHz

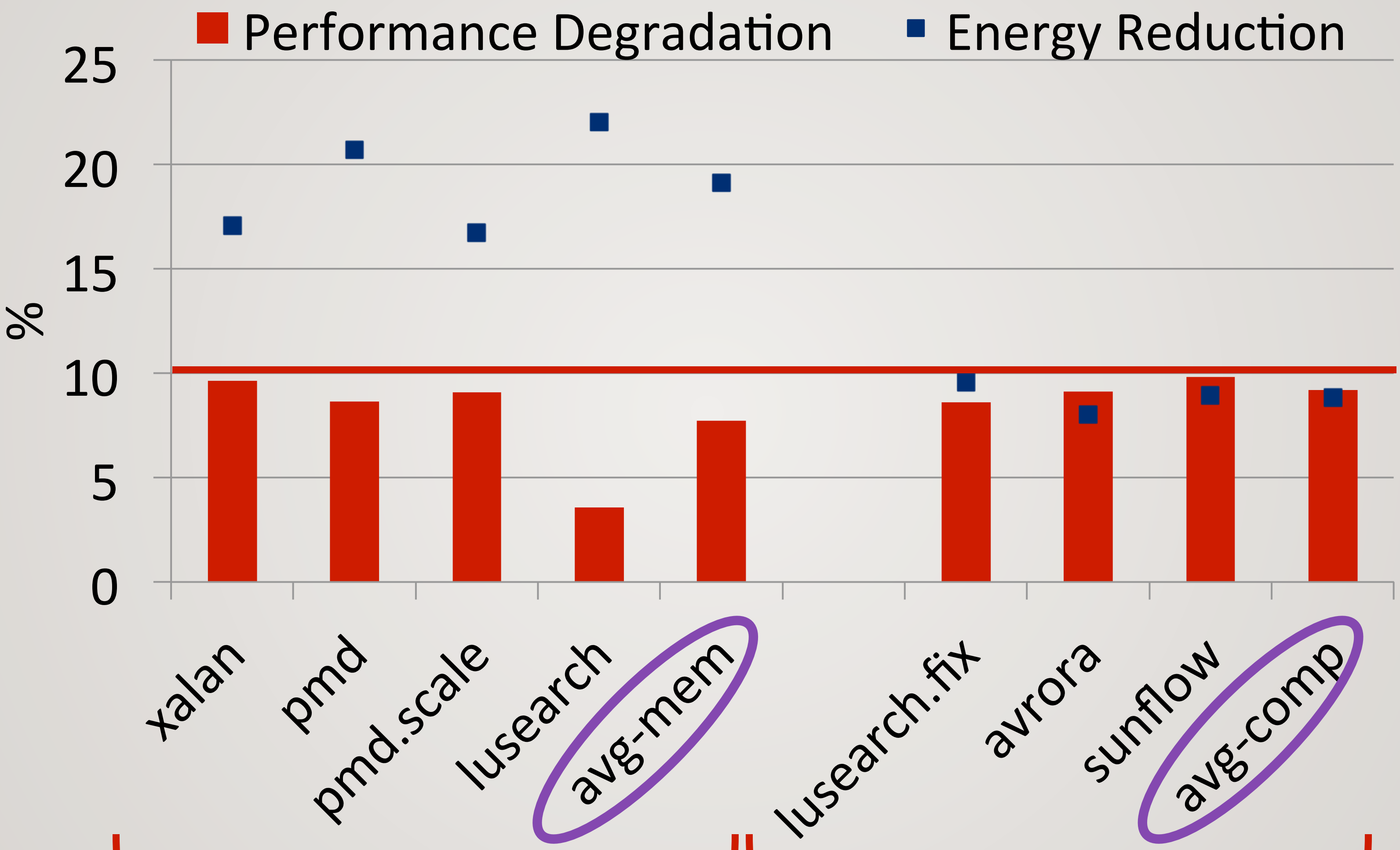
Energy Manager

tolerable_performance_degradation



Quantum
5 ms

Energy Savings



Memory Intensive

Compute Intensive

Conclusions

- DEP+BURST: First predictor that accounts for
  - Application and service threads
  - Synchronization → inter-thread dependencies
  - Store bursts
- High accuracy
  - Less than 10% estimation error for seven Java bmarks.
- Negligible hardware cost
  - One extra performance counter
  - Minor book-keeping across epochs
- Demonstrated energy savings
  - 20 % avg. for a 10% slowdown (mem-intensive Java apps.)

DVFS PERFORMANCE PREDICTION FOR MANAGED MULTITHREADED APPLICATIONS

Thank You !

Shoaib.Akram@elis.UGent.be