

**MATS: MULTICORE
ADAPTIVE TRACE
SELECTION**

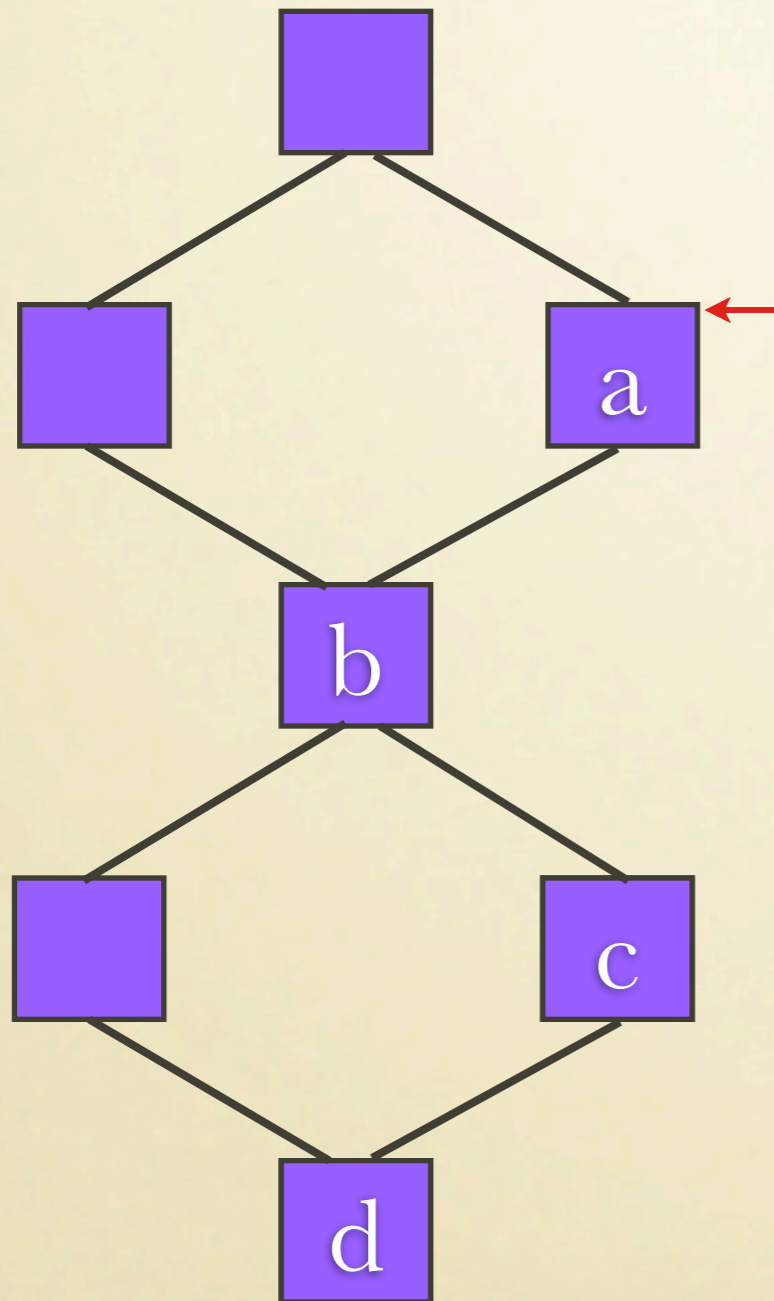
**JASON MARS, MARY LOU SOFFA
UNIVERSITY OF VIRGINIA**

BINARY LEVEL DYNAMIC OPTIMIZATION

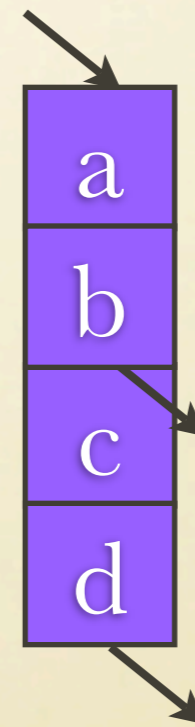
- Optimize Based on Runtime Information
- Optimize Paths of Execution (Traces)
- Must Identify Hot Paths
 - Hot = frequently executed
- “90% Execution in Only 10% of Code”

WHAT IS A HOT TRACE?

Application Control Flow



Hot Trace



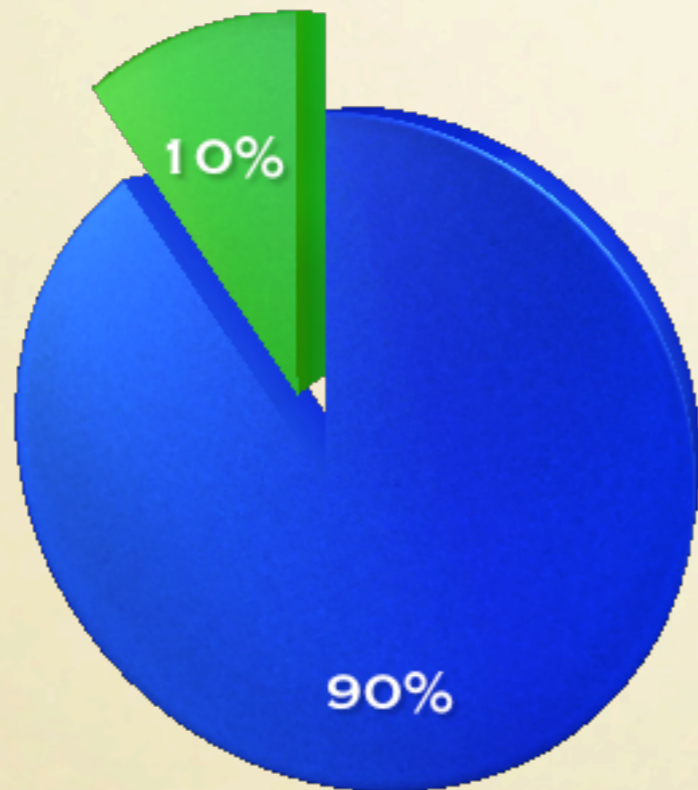
TRACE QUALITY

- Optimization Potential Depends on Trace Quality
 - Time Spent on Traces
 - Trace Length
 - Ideal Trace Executions
 - Trace Executes to Completion

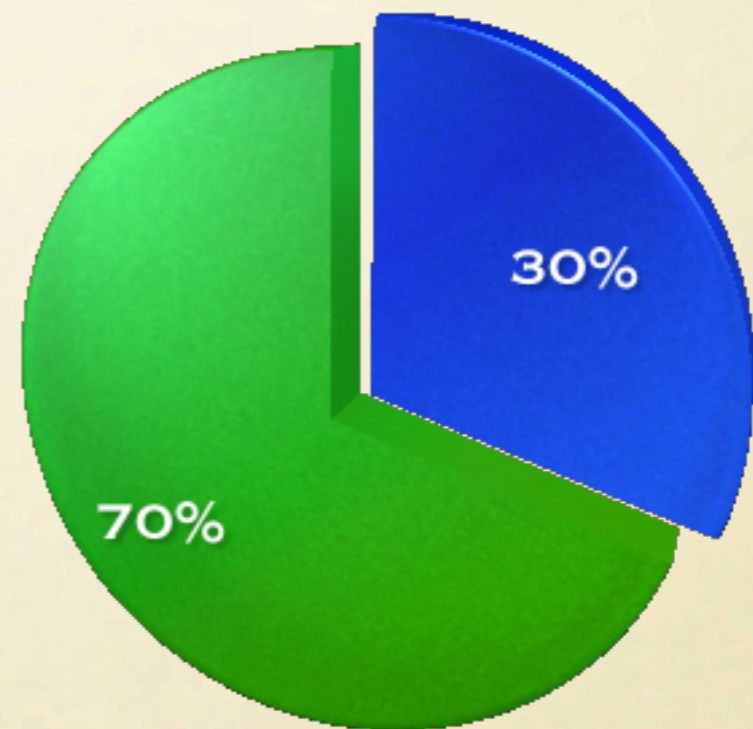
TRACE QUALITY: TIME SPENT ON TRACES

Good

Bad



● ON TRACE ● ORIGINAL CODE

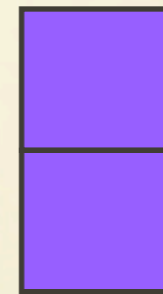
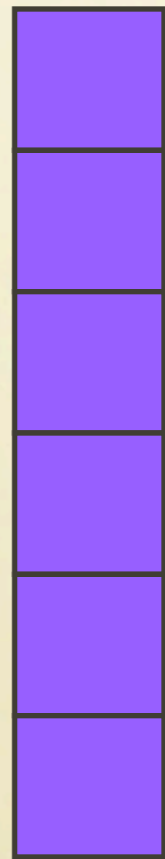


● ON TRACE ● ORIGINAL CODE

TRACE QUALITY: LENGTH

Good

Bad



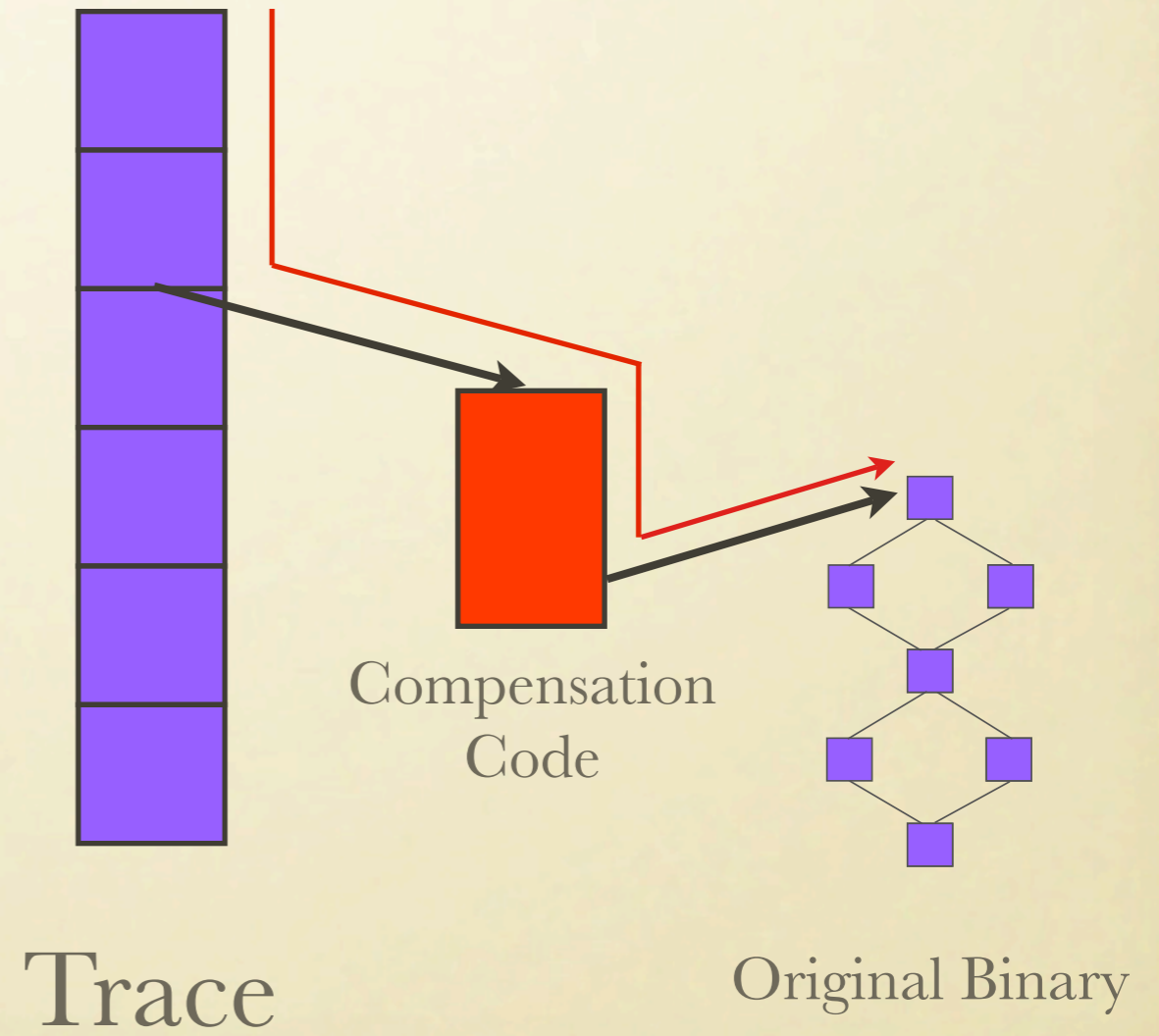
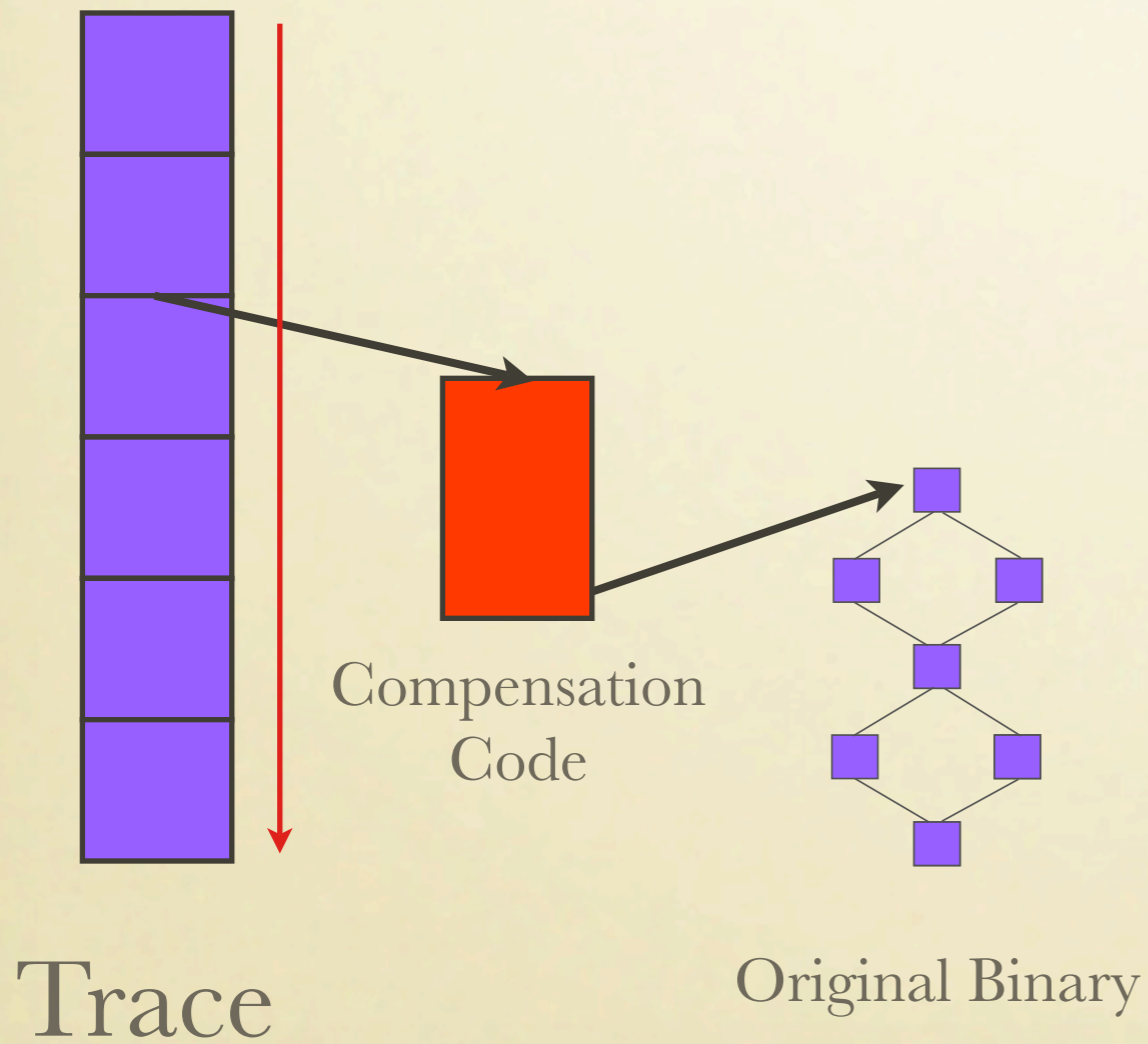
Trace

Trace

TRACE QUALITY: IDEAL TRACE EXECUTION

Good

Bad



GOALS

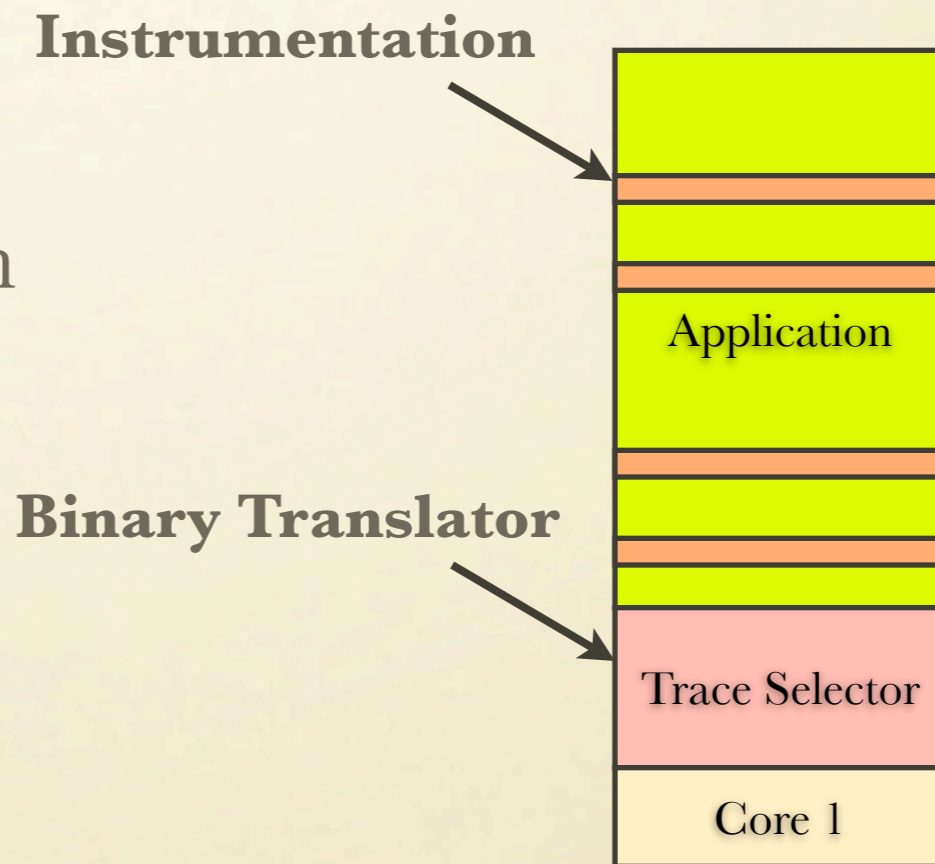
“How can we perform trace selection without incurring too much application overhead?”

“If we use a more expensive algorithm would our traces be better?”

- Low/No Overhead Traces Selection
- Higher Quality Traces
 - Long
 - Hot
 - Ideal

TYPICAL TRACE SELECTION APPROACH: NET

- Dynamo Model
 - Binary Translator Instruments Application
 - Next Executing Tail (NET)
 - Backedge Branches Profiled



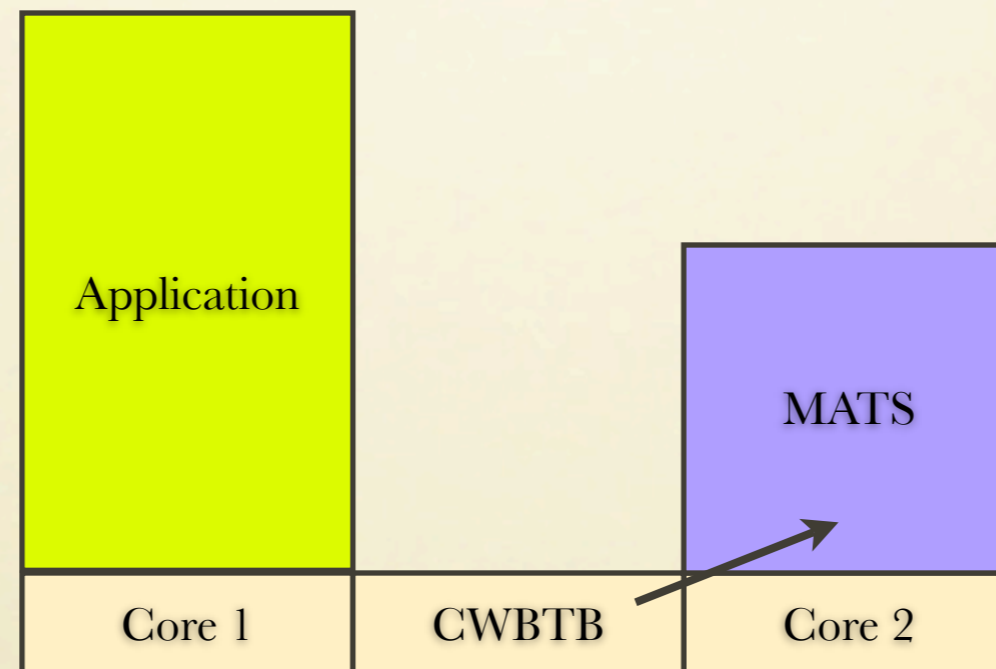
CHALLENGES

- Overhead Due to Application Obtrusion
 - Overhead impacts optimization benefits
- Limitation of Monitoring Analysis Due to the Overhead Constraints
 - Trace Quality May Suffer
 - Potential benefits depends on trace quality

CHALLENGES

Multicore Brings
New Possibilities

MATS APPROACH: OVERVIEW



The Core Wide Branch Trace Buffer (CWBTB) is a centralized BTB that is readable by all cores

GOALS

“How can we do trace selection without incurring too much application overhead?”

“If we use a more expensive algorithm would our traces be better?”

- Low/No Overhead Traces Selection



- Higher Quality Traces

- Long

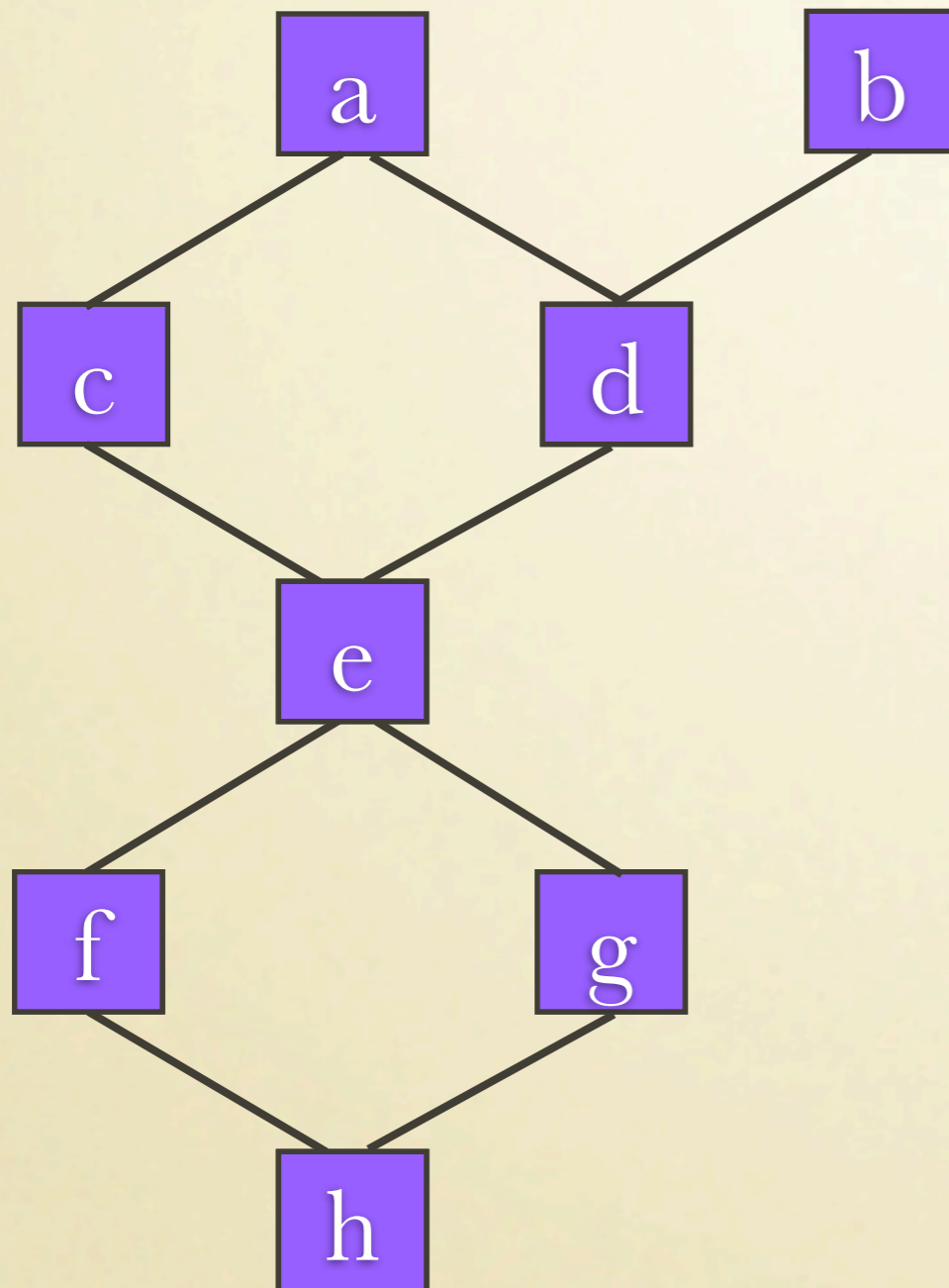
- Hot

- Ideal

MATS APPROACH: TRACE SELECTION ALGORITHM

- Pattern Based Approach
 - Patterns in a sequence of taken branches from CWBTB
 - The Sequitur algorithm used for pattern detection
 - Take advantage of global information

MATS APPROACH: TRACE SELECTION ALGORITHM

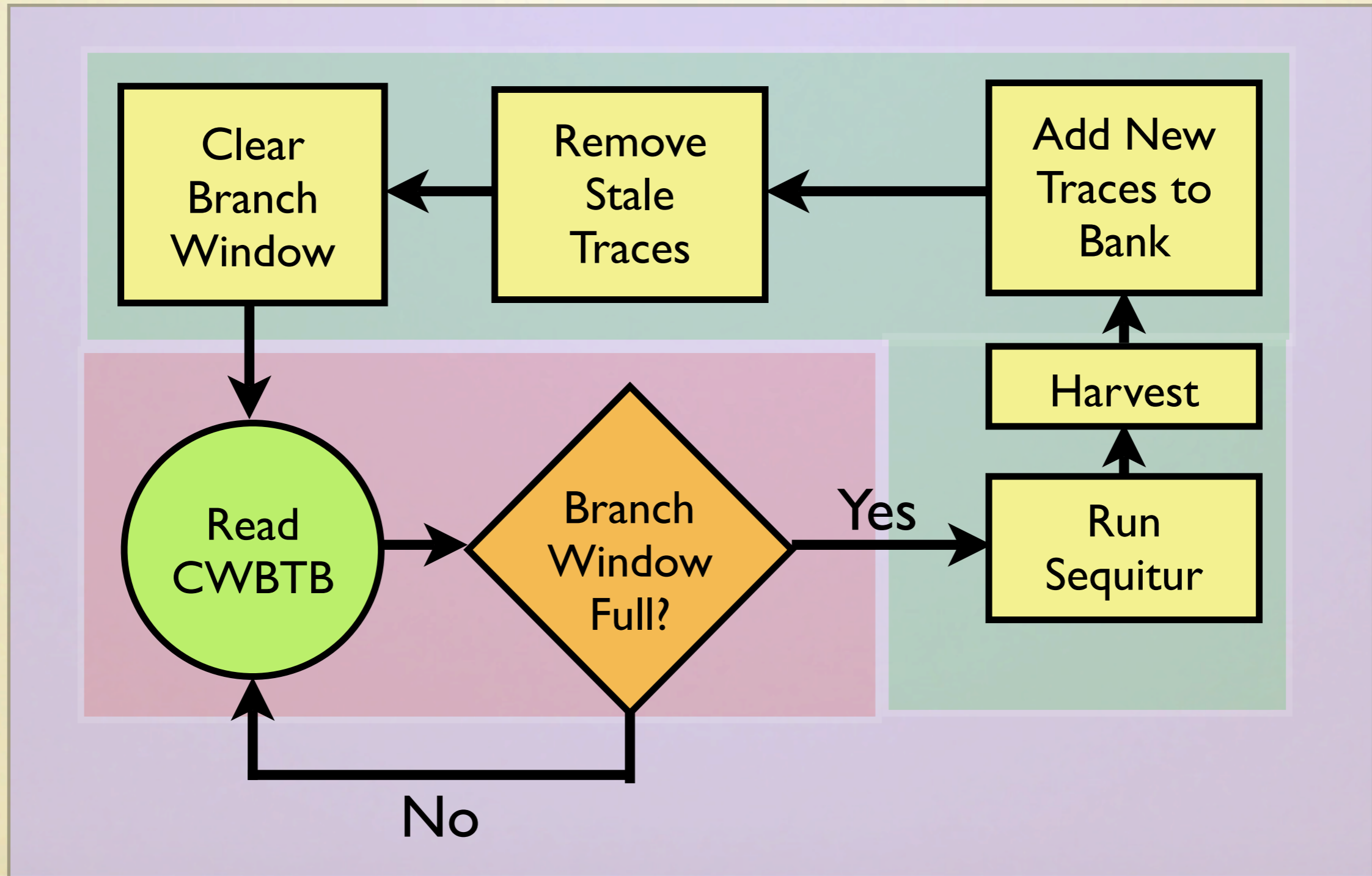


Execution Path:

acefhade**fh**bbdef**h**



MATS APPROACH: SPECULATION ENGINE

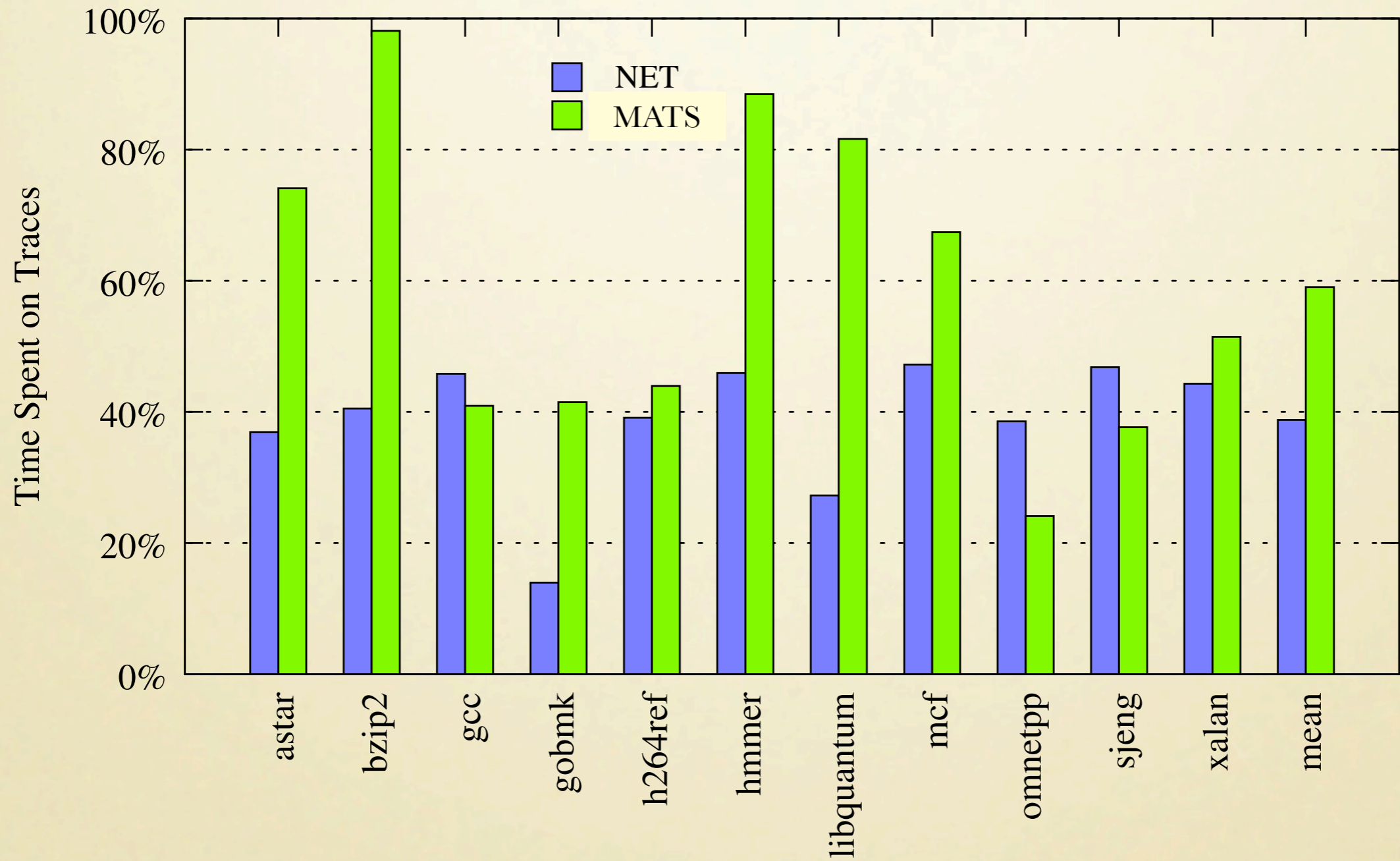


EVALUATION

- Goal: Determine if our traces are better.
 - Time Spent on Traces
 - Ideal Traces Execution
 - Trace Length

RESULTS:

TIME SPENT ON TRACES



GOALS

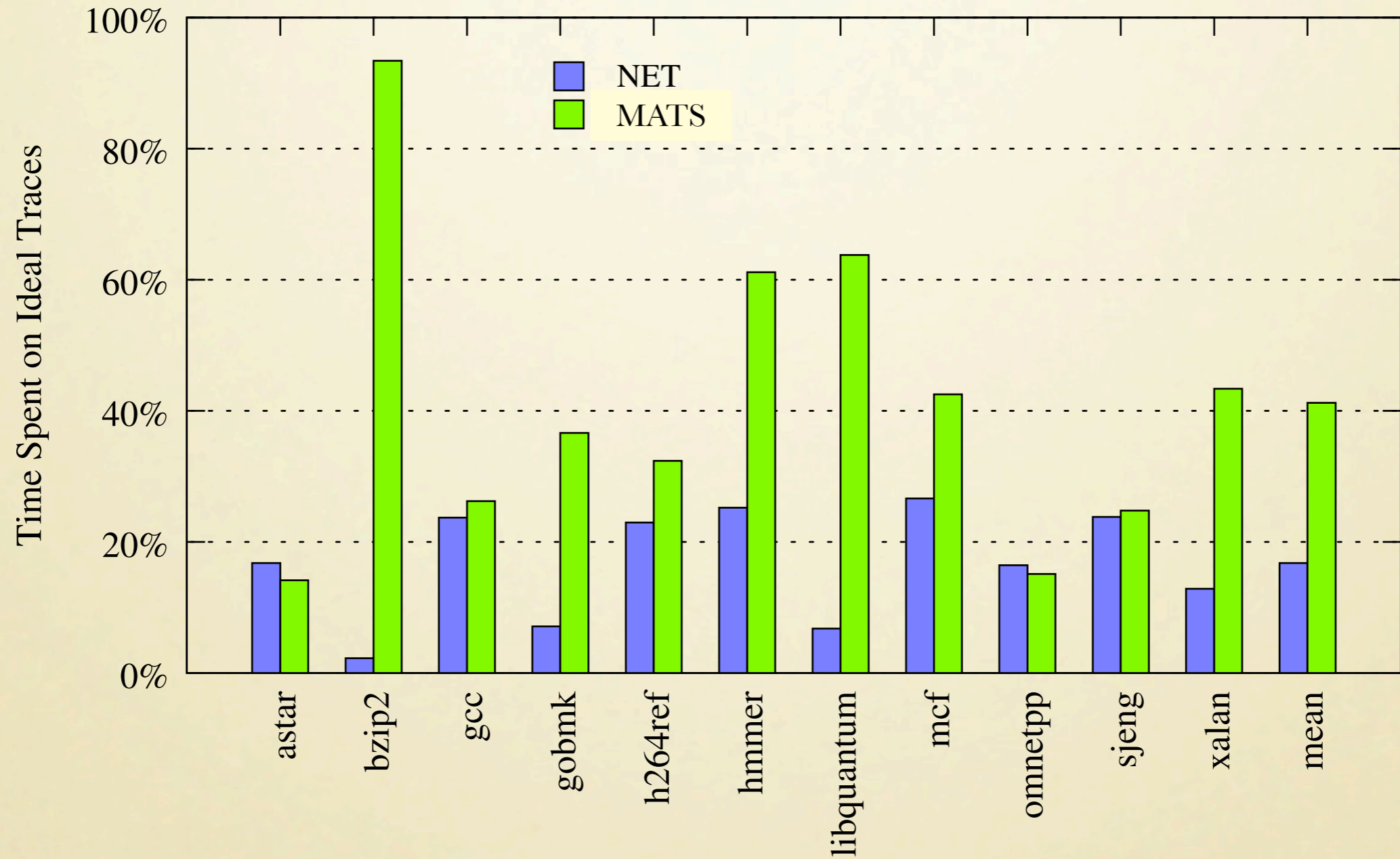
“How can we do trace selection without incurring too much application overhead?”

“If we use a more expensive algorithm would our traces be better?”

- Low/No Overhead Traces Selection 
- Higher Quality Traces
 - Long
 - Hot 
 - Ideal

RESULTS:

TIME ON IDEAL TRACES



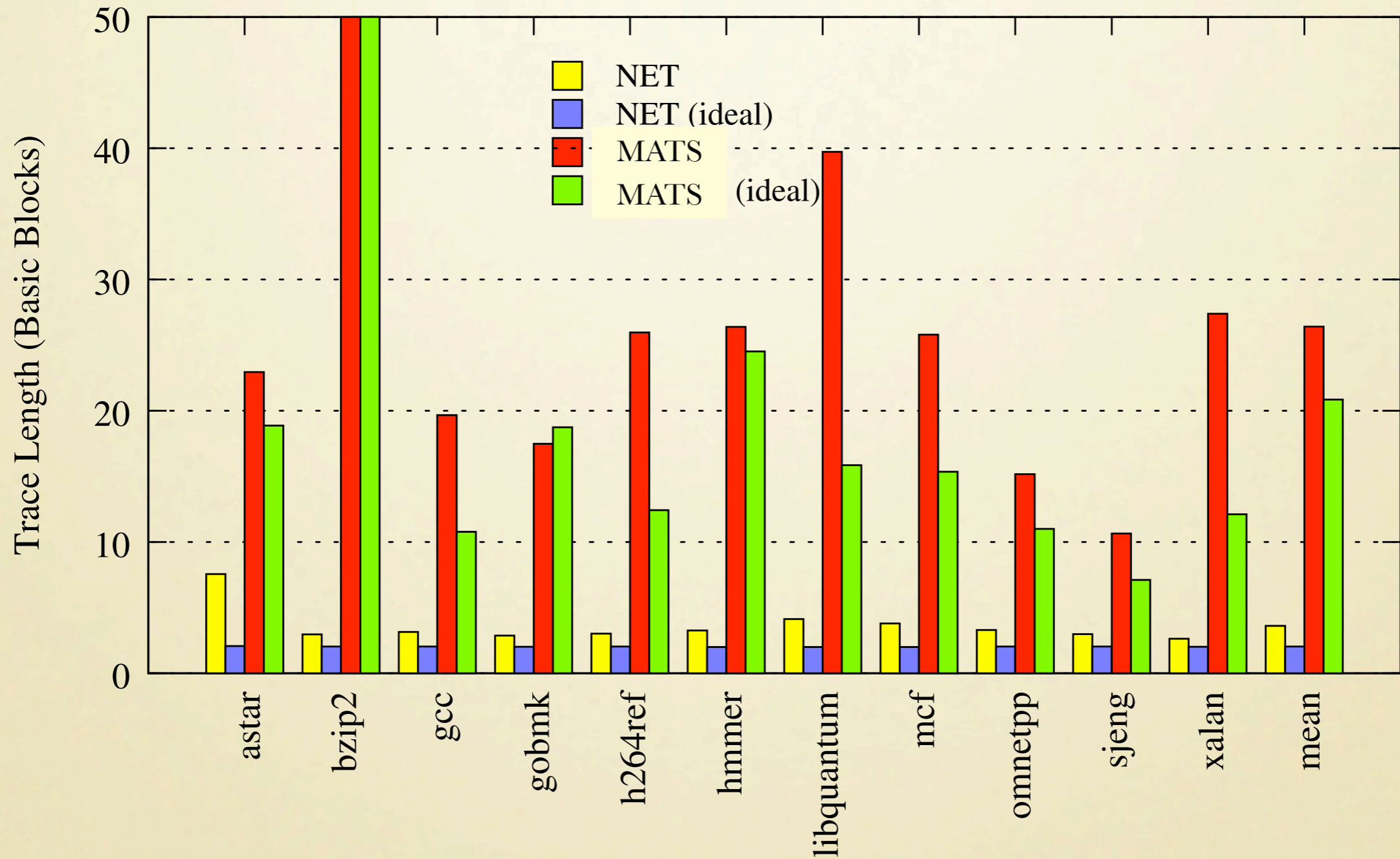
GOALS

“How can we do trace selection without incurring too much application overhead?”

“If we use a more expensive algorithm would our traces be better?”

- Low/No Overhead Traces Selection
- Higher Quality Traces
 - Long
 - Hot
 - Ideal

RESULTS: TRACE LENGTH



GOALS

“How can we do trace selection without incurring too much application overhead?”

“If we use a more expensive algorithm would our traces be better?”

- Low/No Overhead Traces Selection
- Higher Quality Traces
 - Long
 - Hot
 - Ideal

SUMMARY

- Used multicore architecture to unobtrusively perform trace selection
- Designed new trace selection analysis that produces higher quality traces
- Demonstrated that using more expensive trace analysis can produce better traces
- Demonstrated the benefits possible from leveraging performance monitoring hardware

FUTURE WORK

- Study the best “harvesting” criteria for patterns, and the best trace invalidation criteria and how they apply to different workloads
- Develop a self tuning speculation engine to tune to different applications and application phases
- Integrate this trace selector with an optimizing engine

RELATED WORK

- Hardware Approach
 - Trace Cache
 - *does not facilitate software dynamic optimization*
 - Trident (Zhang et al.)
 - *not available*

RELATED WORK

- Software Dynamic Approach
 - Employed successfully by :
 - *Dynamo (Bala, Duesterwald et al.)*
 - Sequitur Used for Static Path Profiling
 - *Ball and Laurus*
 - *See Paper for more*



QUESTIONS?

DISCUSSION QUESTIONS

- With all this Multi / Manycore talk going on, what can we do for single threaded and legacy programs?
- Is there a place of Binary Translator VEE Research for the Multicore?
- Is a Managed OS (like Singularity) The Future, or Just a waste of time?
(Singularity = OS built around CLR)

SEQUITUR

Sequitur: $S \rightarrow A -8 0 A$
 $A \rightarrow 12 -22$

Tree:

