

MultiCore Expo

April 2, 2008



# Multicore Benchmarks Help Match Programming to Processor Architecture

Shay Gal-On  
Director of Software Engineering  
EEMBC

[www.eembc.org](http://www.eembc.org)

# What's Up?

- What is EEMBC
- Multicore benchmarking framework
- Multicore sample results

# Our Focus

- Industry standard benchmarks since 1997
- New: Evaluating current and future development of MP platforms
  - Uncovering MP bottlenecks
  - Analysis of multicore systems

# MultiBench – Overview

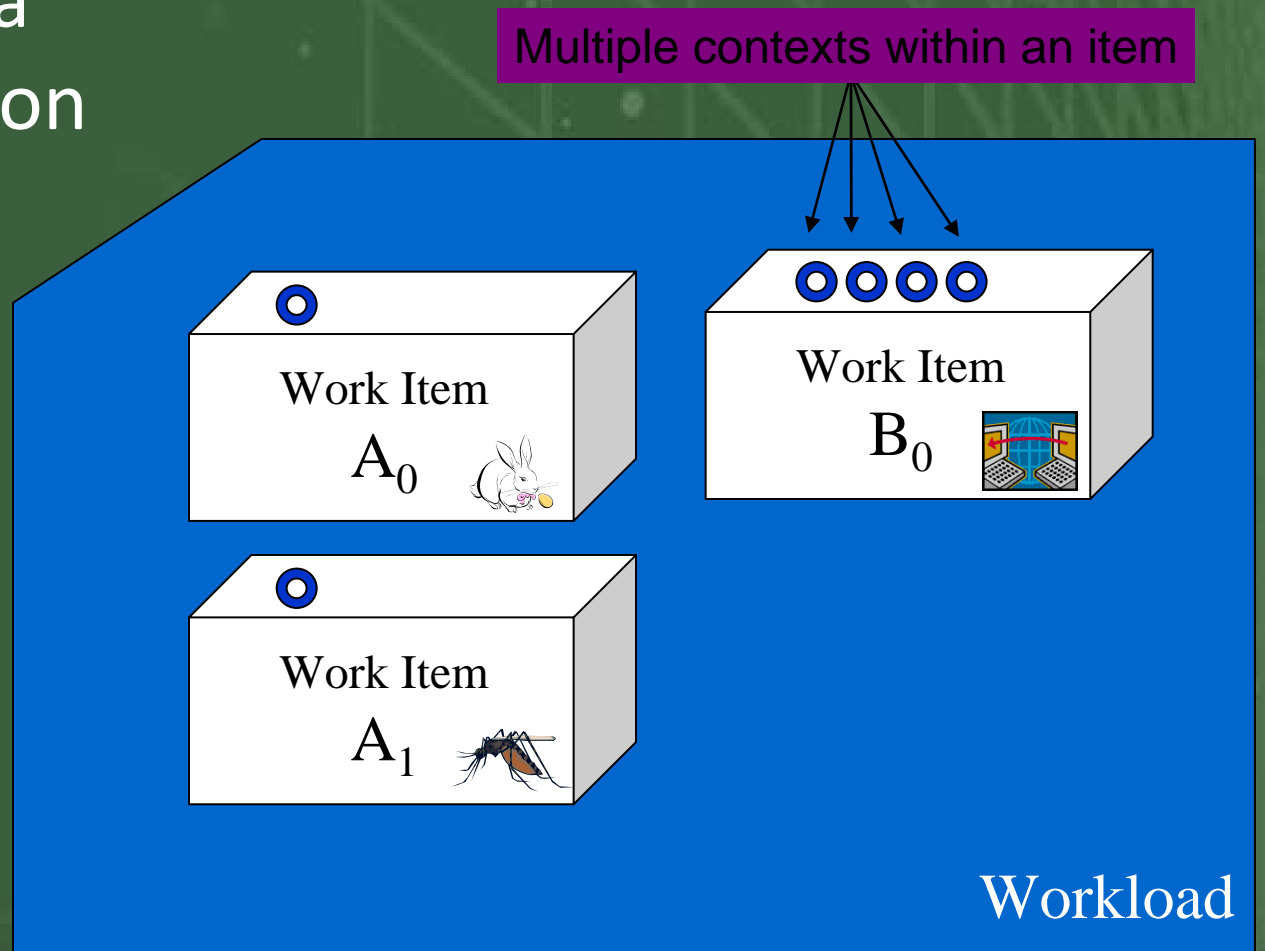
- Multicore is everywhere
  - Hence our focus on benchmarking embedded multicore solutions.
- Initial implementation targets SMP
  - Easiest model to work with
  - Current metrics misleading (rate, DMIPS, etc)
- Workloads and work items
  - Develop workloads closer to real life  
Individual kernels are work items within a workload

# MultiBench – Abstraction

- Easy to run
  - Only 13 calls to implement means quick porting to any platform/OS/toolchain
- Wide range of applications
  - Sufficient functionality with minimum porting effort for most embedded applications.
- Most EEMBC benchmarks ported
  - Thread safety
  - New datasets
  - Common API

# Workloads and Work Items

- Multiple algorithms
- Multiple data
- Decomposition







# Concurrency

Multiple Work Items





Note: Algorithm same, different datasets

Workload

Work Item 1 	Work Item 2 
	Work Item 4 
Work Item 3	

Can be run one item at a time .....

Workload

Work Item 1 	Work Item 2 
	Work Item 4 
Work Item 3	

Or concurrently, depending on processing resources and scheduling

# Multiple Iterations

Example with two contexts doing four repeats

Core 1



Core 2



*Idle Time*  
Start next iteration here

**NOTE:**

Alternatively, the OS may schedule Core 1 and Core 2 to run only hamburgers and cucumbers, respectively. Other combinations are possible.

Time Sequence →



# Work Items and Workers

A collection of threads working on the same item are referred to as workers



# Workload set #1

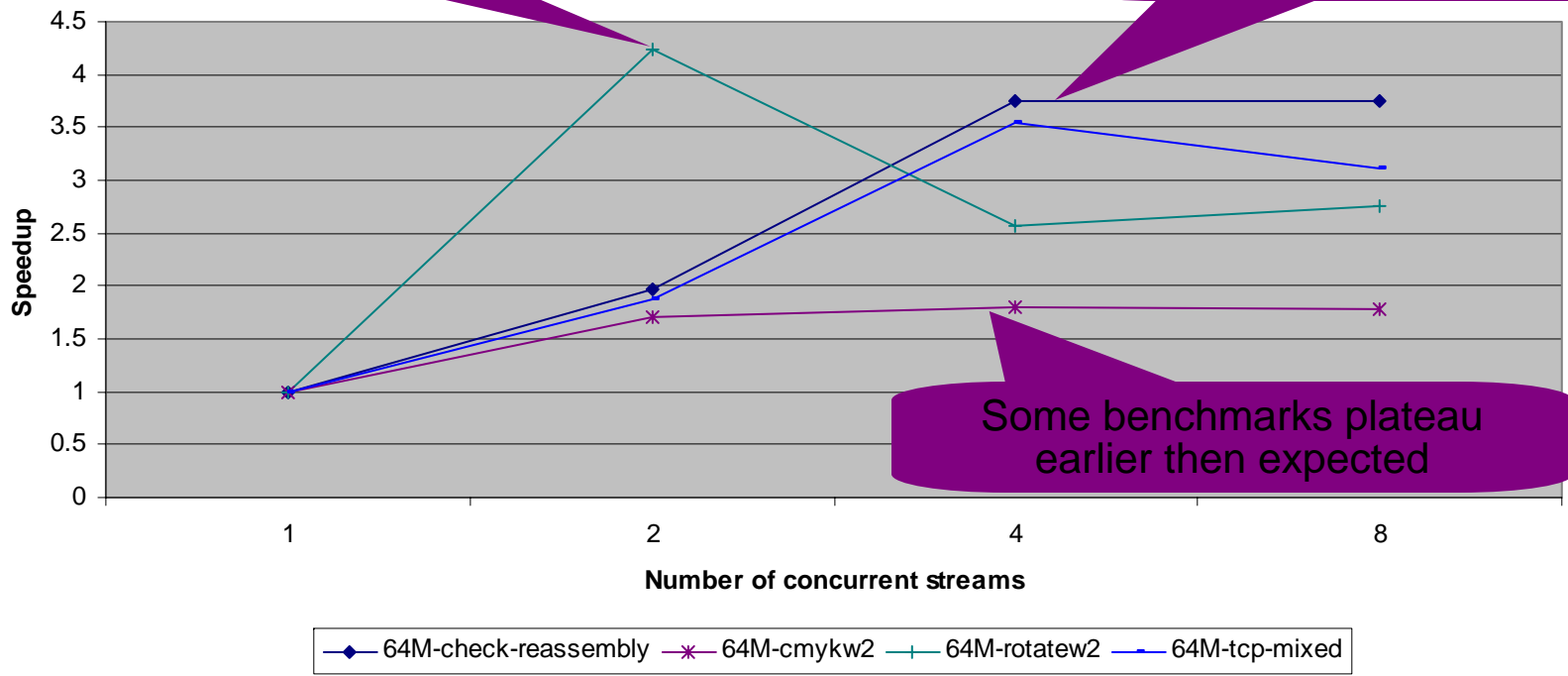
- Carefully selected subset
  - Only 30 workloads over a few kernels
- Benchmark efficiency of various multi-processing system related effects
  - Computation / Synchronization
  - Memory / Cache
- Work items from multiple segments
  - Networking
  - Consumer

# Workloads scale in mysterious ways...

Huge drop in performance when oversubscribed

Quad Core

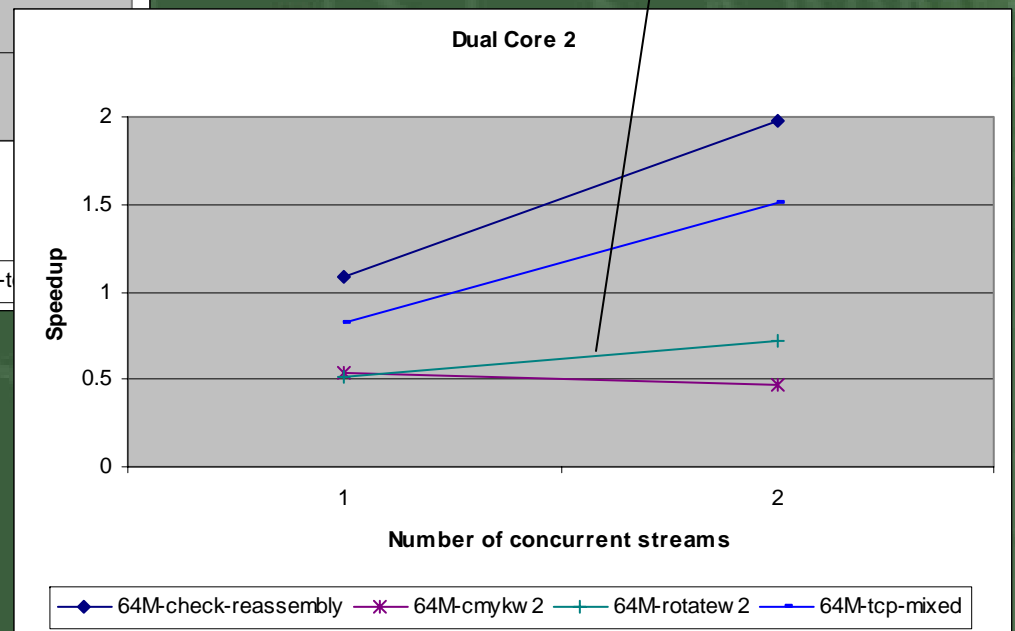
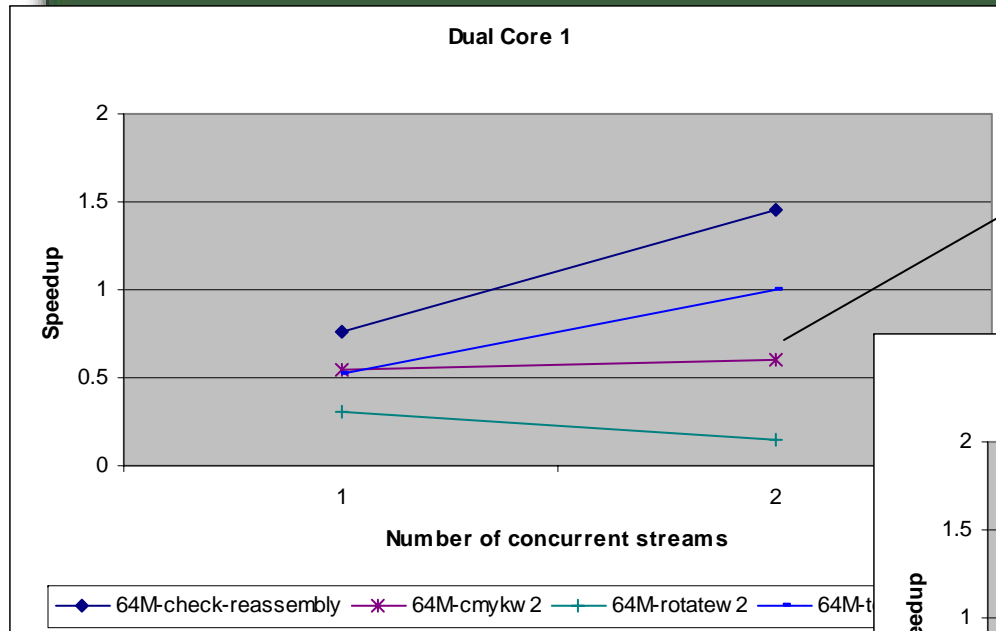
Nice scaling on networking only workloads



Some benchmarks plateau earlier than expected

# Compare Two Systems

Various workloads expose different trends in systems

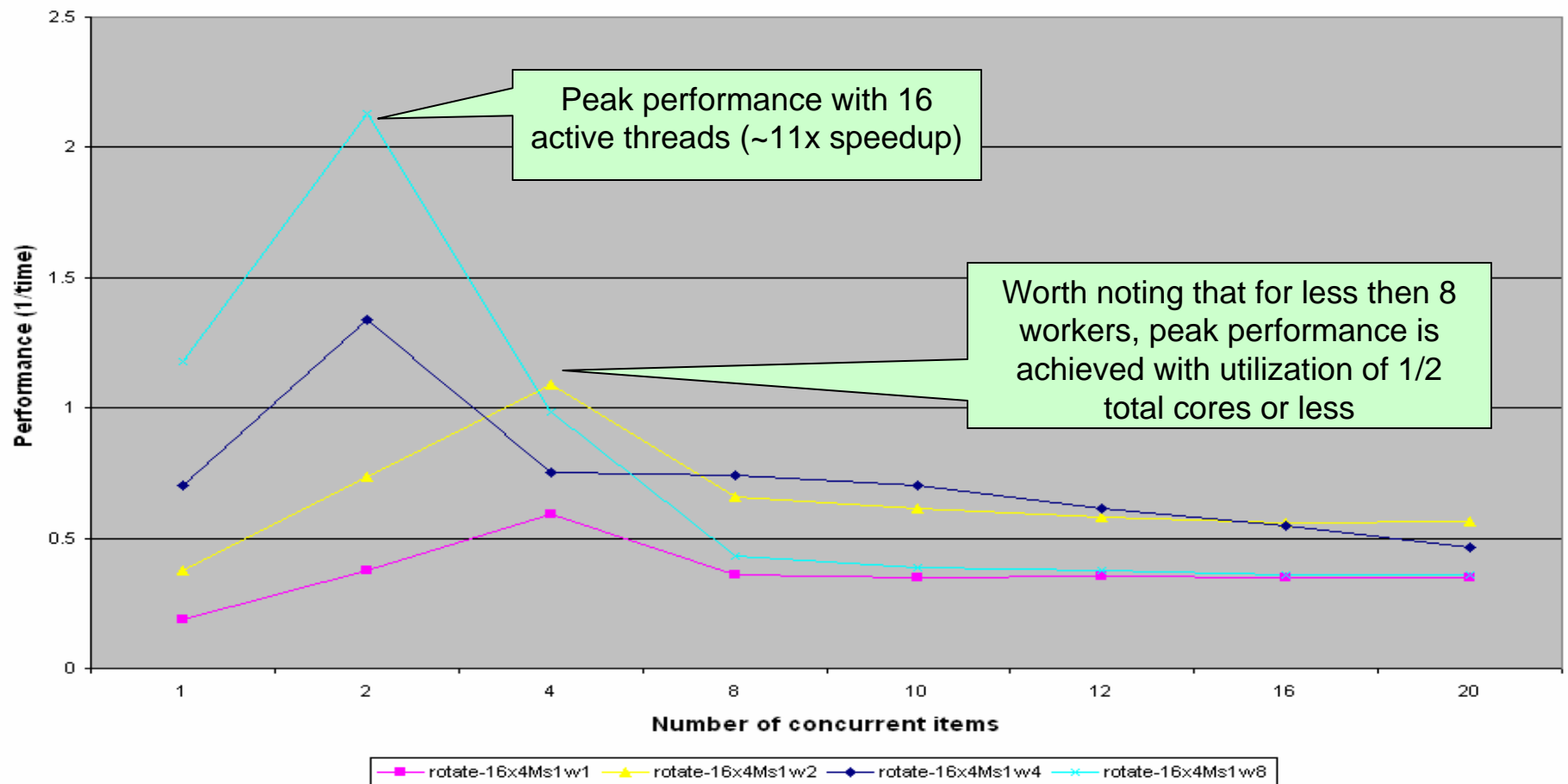


# Image Rotation, 90° CW

- Simple algorithm, very little computation
- Easy to slice to workers, since each pixel is independent.
- Depending on slicing and image size, can exercise the system in interesting ways.
- Many other applications use similar data movement patterns.

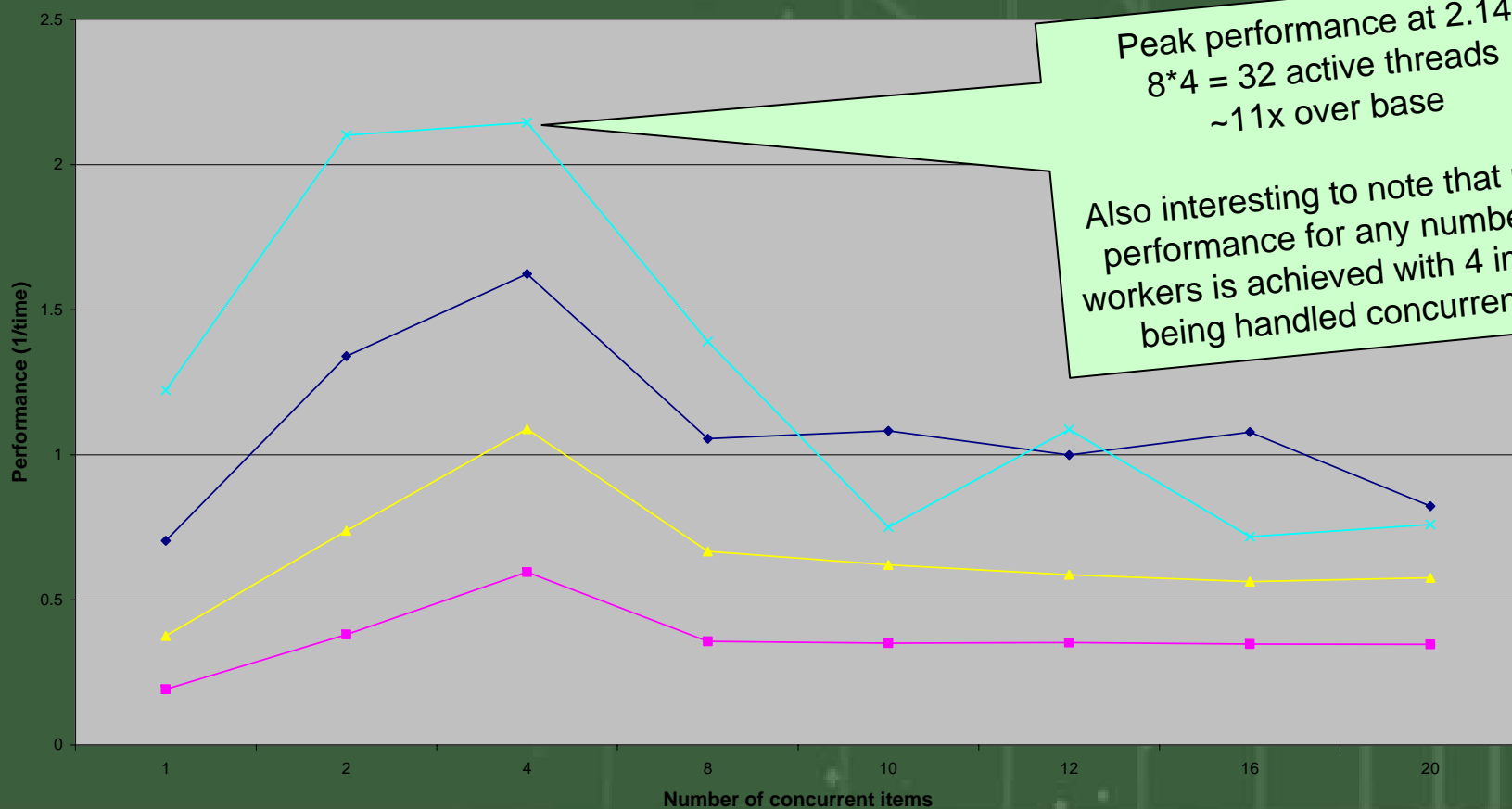
# Rotation Benchmark (Slice = 1)

Performance Scaling



# Rotation Benchmark (Slice = 4)

Performance Scaling



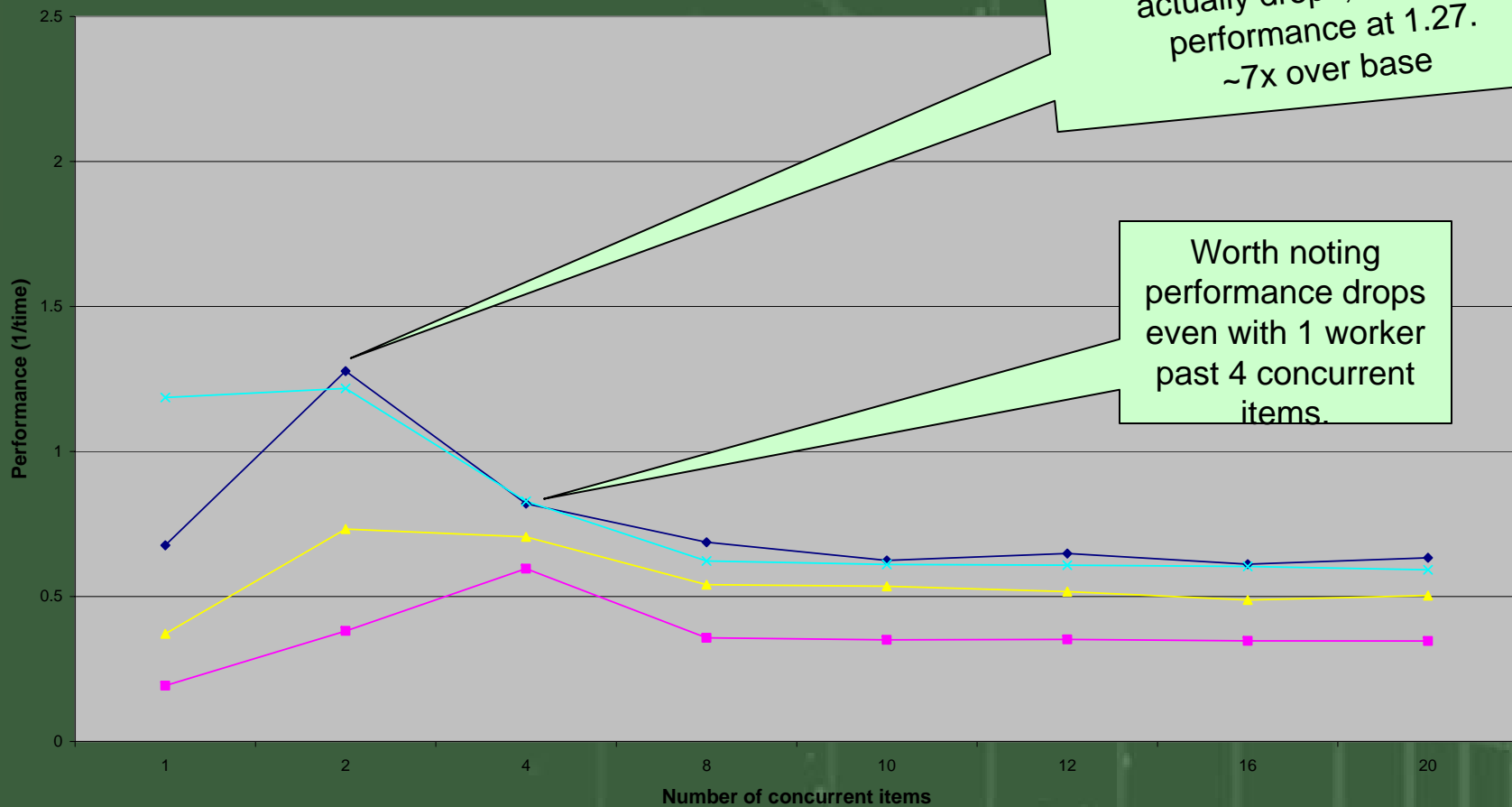
Peak performance at 2.14.  
 $8 * 4 = 32$  active threads  
~11x over base

Also interesting to note that peak performance for any number of workers is achieved with 4 images being handled concurrently

rotate-16x4Ms4w1 rotate-16x4Ms4w2 rotate-16x4Ms4w4 rotate-16x4Ms4w8

# Rotation Benchmark (Slice = 32)

Performance Scaling



With larger slices, performance actually drops, with peak performance at 1.27. ~7x over base

Worth noting performance drops even with 1 worker past 4 concurrent items.

rotate-16x4Ms32w1 rotate-16x4Ms32w2 rotate-16x4Ms32w4 rotate-16x4Ms32w8



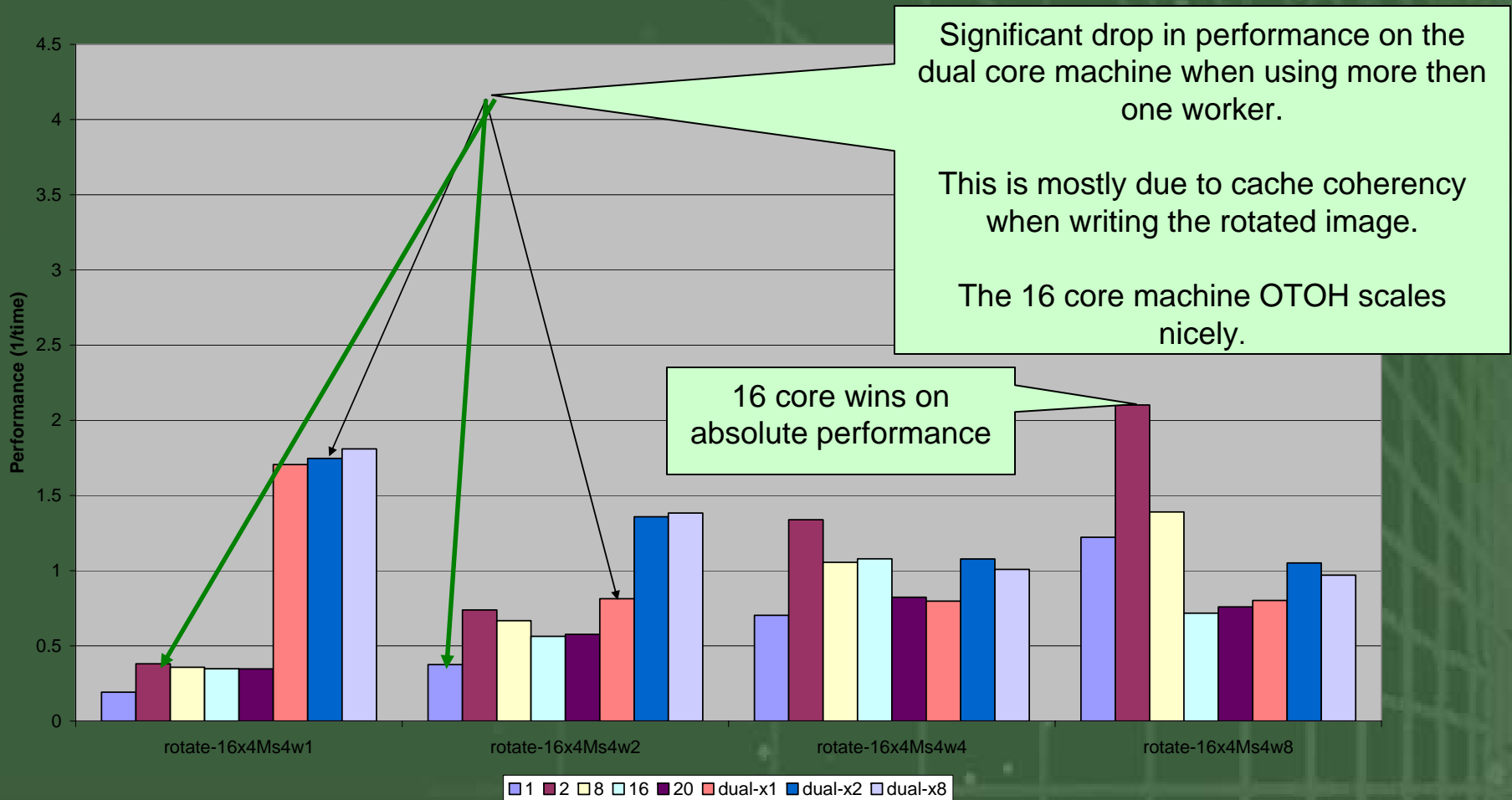
# Use results to design software!

- Results show low overhead for sync
  - Take advantage of data decomposition.
  - Use medium to small granularity.
- Results show throughput max with small number of concurrent data streams
  - Pay more attention to the lower level concurrency.
- Results show bottlenecks
  - Make sure system resources are not oversubscribed by testing with lower load.

# Rotation Benchmark Comparison

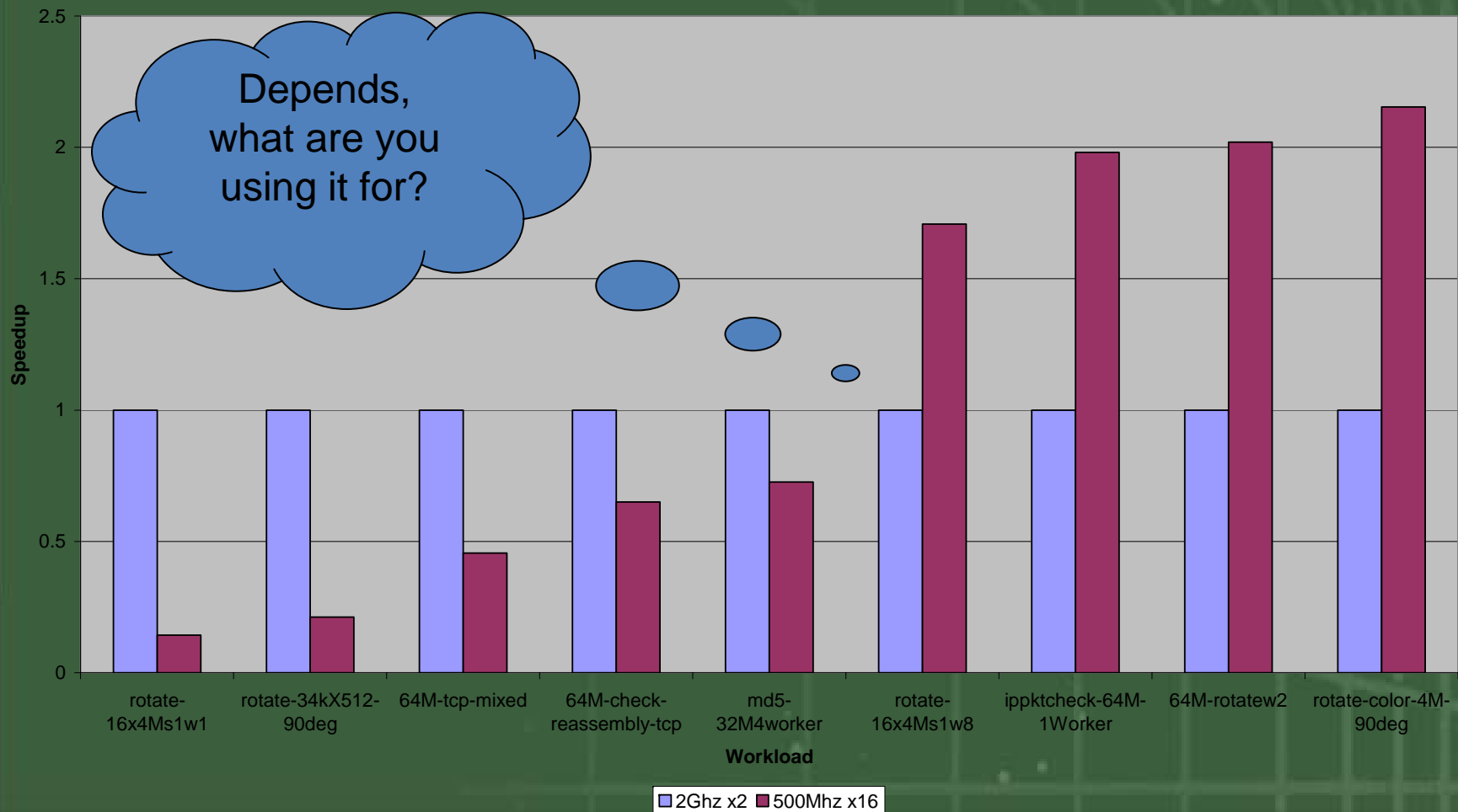
## Dual 2GHz vs. 16 core at 500MHz

Performance Comparison, Rotation, 2x2G vs 16x500MHz



# So which system is better?

Comparing max performance of powerful 2x2G with simple 16x500MHz



# Composing Workloads

The screenshot displays the MITH Workload Creator application. The interface includes a menu bar (File, Help), a filter input field containing 'jp', and a 'Visualize' button. The 'Workload Name' is 'new\_workload', 'Iterations' is 10, and 'Min Contexts' is 1. A 'Workload Description' field is present but empty. Below this, there are two main panels: 'Available Items' and 'Workload Definition'. The 'Available Items' panel contains a table of items, and the 'Workload Definition' panel contains a table of work items with their connections. A 'Show' dialog box is open, displaying a dependency graph with four nodes and their connections.

Item	Index	Data
cjpeg-data1	0	data1
cjpeg-data2	1	data2
cjpeg-data3	2	data3
cjpeg-data4	3	data4
cjpeg-data5	4	data5
cjpeg-data6	5	data6
cjpeg-data7	6	data7
djpeg-data1	0	data1
djpeg-data2	1	data2
djpeg-data3	2	data3
djpeg-data4	3	data4
djpeg-data5	4	data5
djpeg-data6	5	data6
djpeg-data7	6	data7

Work Item	Index	Data	Connections
aiffr	3	32M1worker	1:data,
bitmnp	3	data32M	
cjpeg-data1	0	data1	3:data,3:ctrl,
djpeg-data1	0	data1	

**Show** dialog box content:

```
graph TD; A("0:aiffr") ---|data| B("1:bitmnp"); C("2:cjpeg-data1") ---|data| D("3:djpeg-data1"); C ---|ctrl| D;
```

# EEMBC Benchmarking Services

- Analysis of complex systems
  - Take advantage of in-house expertise
  - Quickly pick the right processor
- Targeted benchmarking
  - Creation of relevant benchmarks
  - Creation of automated benchmark systems
- Quick benchmark results
  - Utilize expertise with multiple embedded environments
  - Utilize experience with EEMBC suites and other benchmarks
- Guidance for optimizing your applications

## Summary

- EEMBC has a new suite for burning multicore issue.
- Use EEMBC benchmarks to guide software development as well as analyze platforms.
- EEMBC launched new service to assist with benchmarking and analysis.



**Questions?**