

Getting OpenMP Up To Speed

Ruud van der Pas

Distinguished Engineer

SPARC Microelectronics

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

ORACLE

Santa Clara, CA, USA

SC'15 Talk at OpenMP Booth

Wednesday, November 18, 2015

“OpenMP Does Not Scale”

Ruud van der Pas

Distinguished Engineer

SPARC Microelectronics

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

ORACLE

Santa Clara, CA, USA

SC'15 Talk at OpenMP Booth

Wednesday, November 18, 2015

“OpenMP Does Not Scale”



A Common Misconception

But A Programming Model Can Not “Not Scale”

What Can Not Scale:

The Implementation

The System Versus The Resource Requirements

Or You

Top 10 Of Dumb Things To Do



Don't Use OpenMP

Start With A Serial Code That Performs Badly

Don't Use A Profiling Tool

Excessive Use Of Parallel Regions

Excessive Use Of Shared Data

Top 10 Of Dumb Things To Do



Don't Use The Nowait Clause

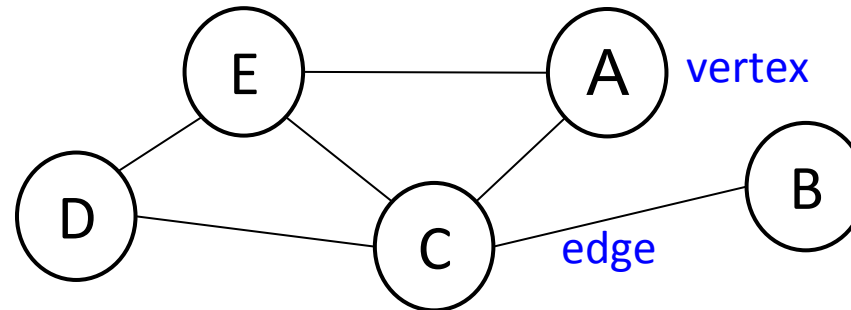
Use Locks Where They Don't Make Sense

Ignore Load Balancing Issues

Ignore cc-NUMA

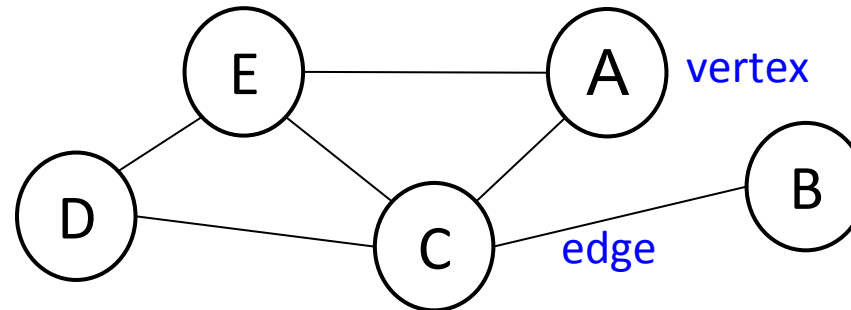
Forget All Of The Above

Graphs In A Nutshell/1



- A graph consists of a set of **vertices** “V” and **edges** “E”
- The vertices “V” are the objects of interest (e.g. people)
- The edges “E” describe a relationship between vertices
 - For example “Do these two people know each other ?”

Graphs In A Nutshell/2



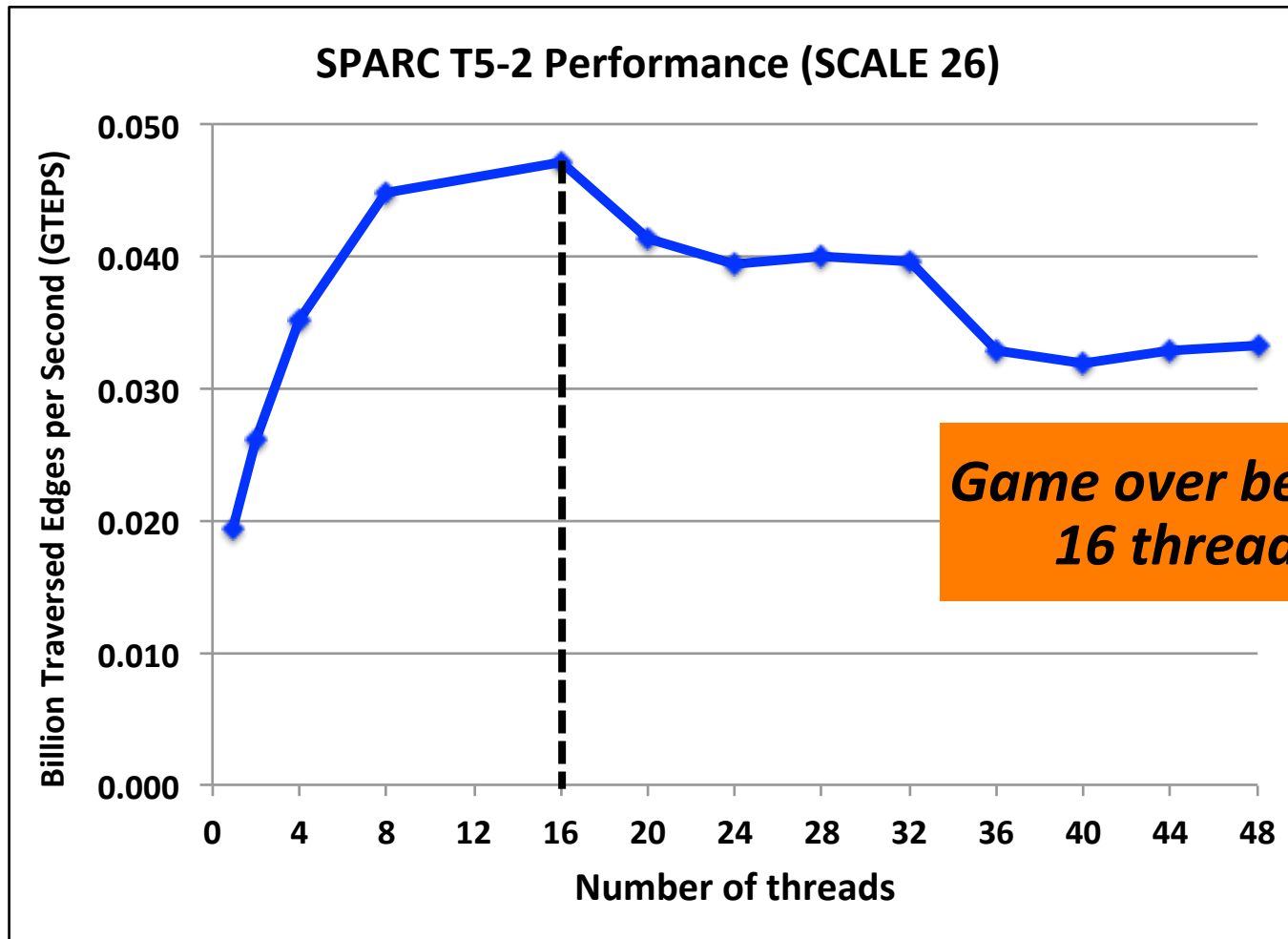
- A key operation on a graph is to **search** for connections
- The performance is measured by the **numbers of edges traversed per second (TEPS)**

The Graph Analysis Benchmark



- Written in C, parallelized with OpenMP
- The key input parameter is called “**scale**”
- The number of vertices is 2^{scale}
- The number of edges is 16 per vertex

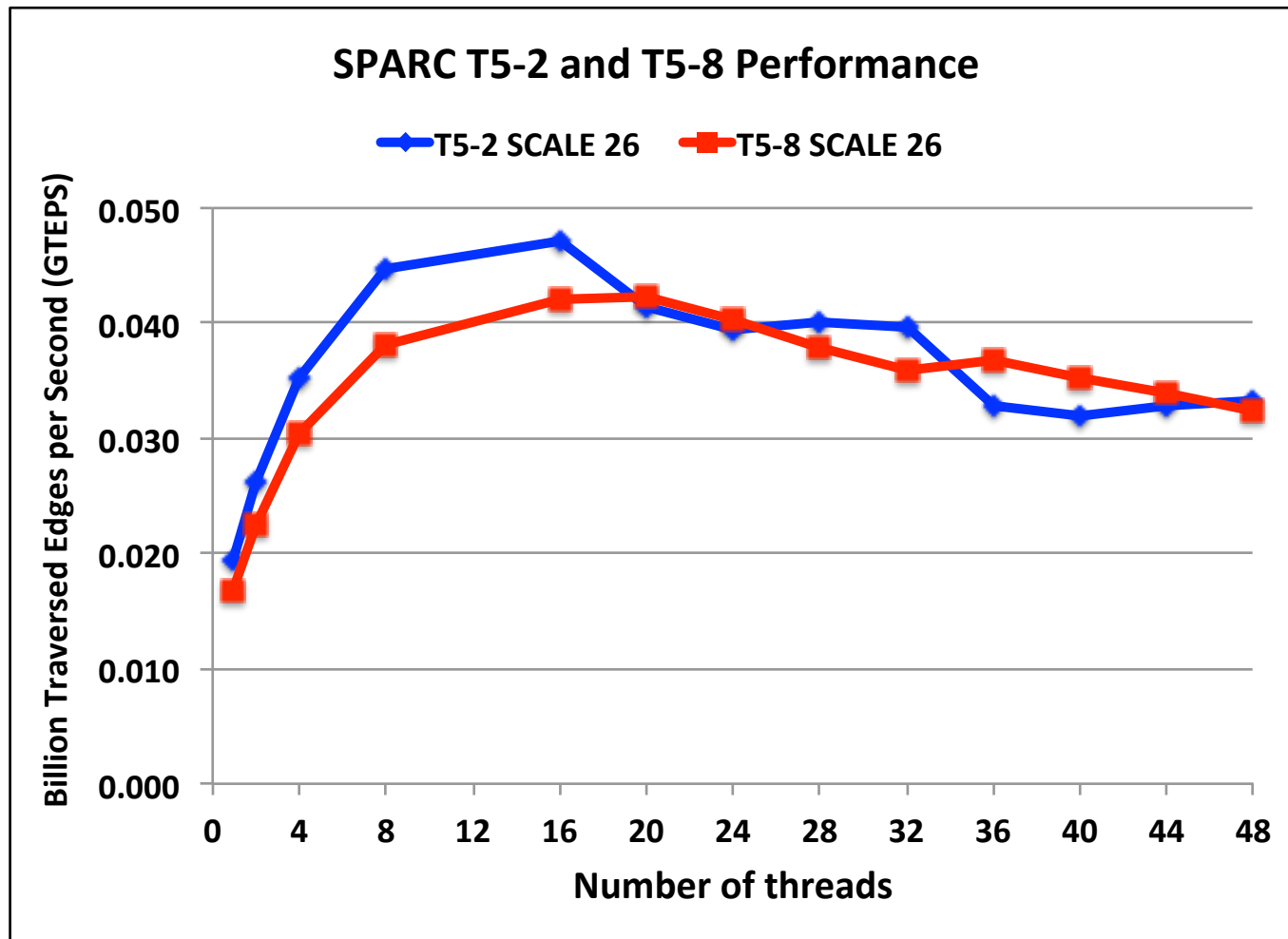
The Initial Performance (35 GB)



That doesn't scale very well

Let's use a bigger machine !

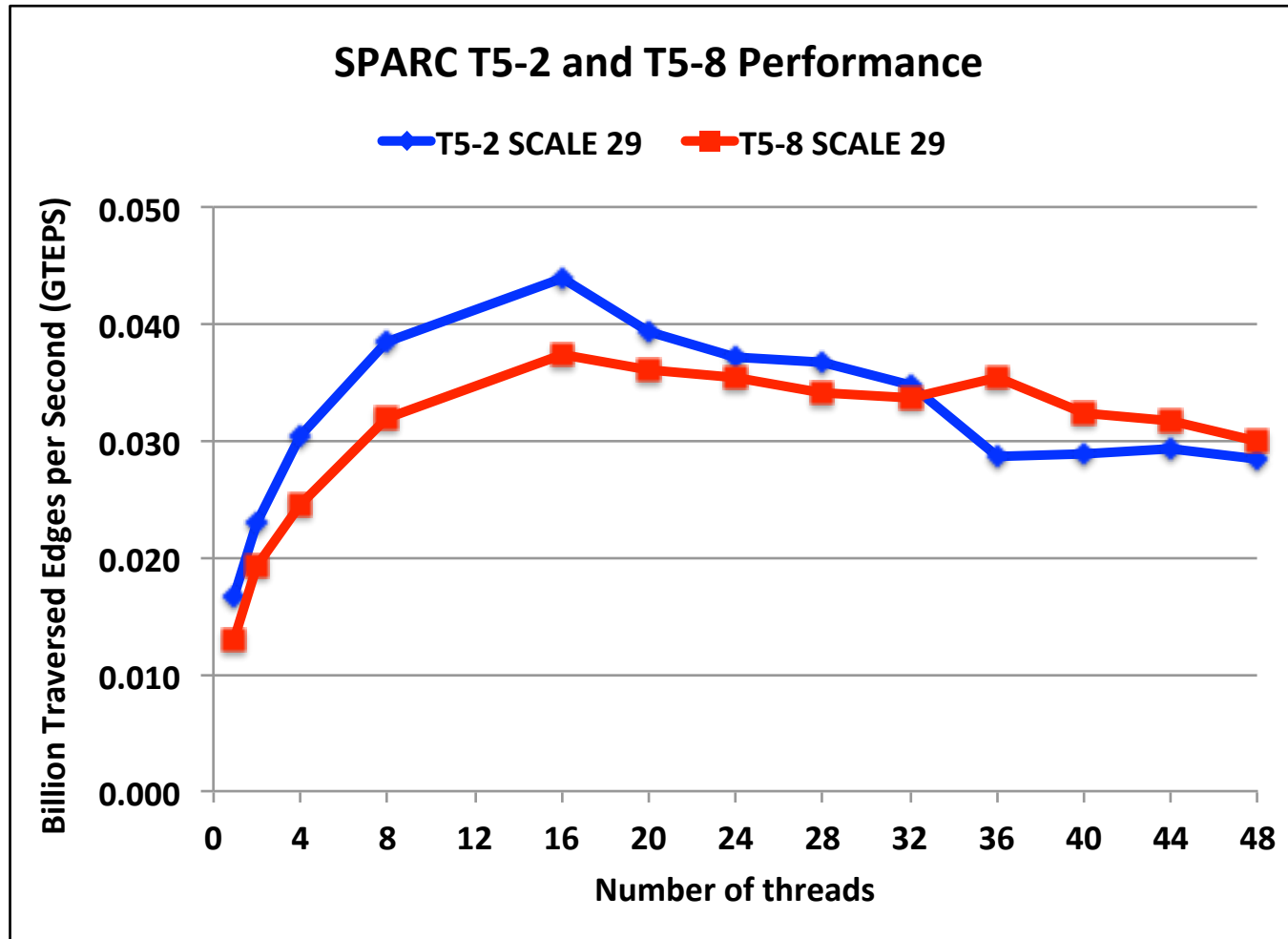
Initial Performance (35 GB)



Oops! That can't be true

Let's run a larger graph !

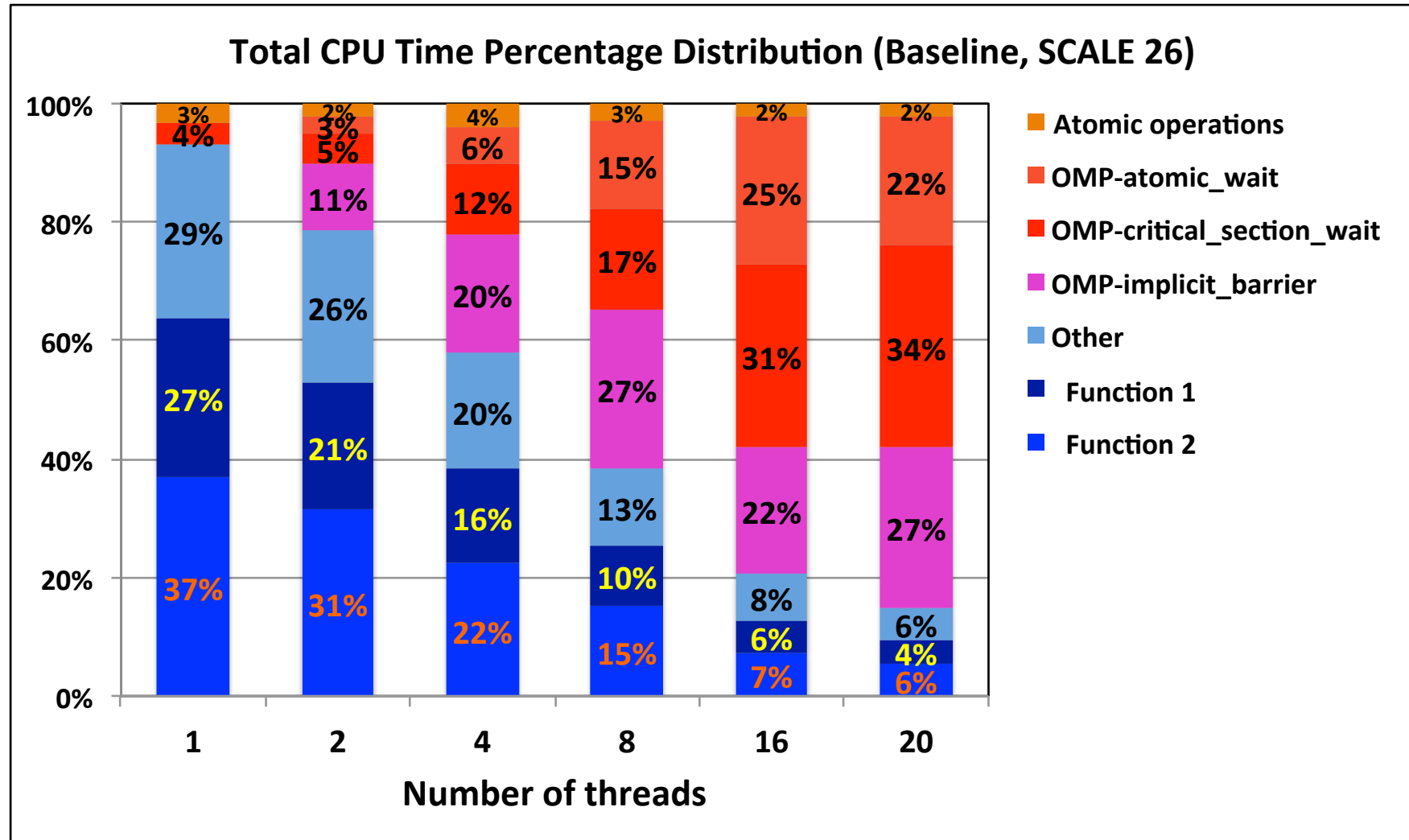
Initial Performance (280 GB)



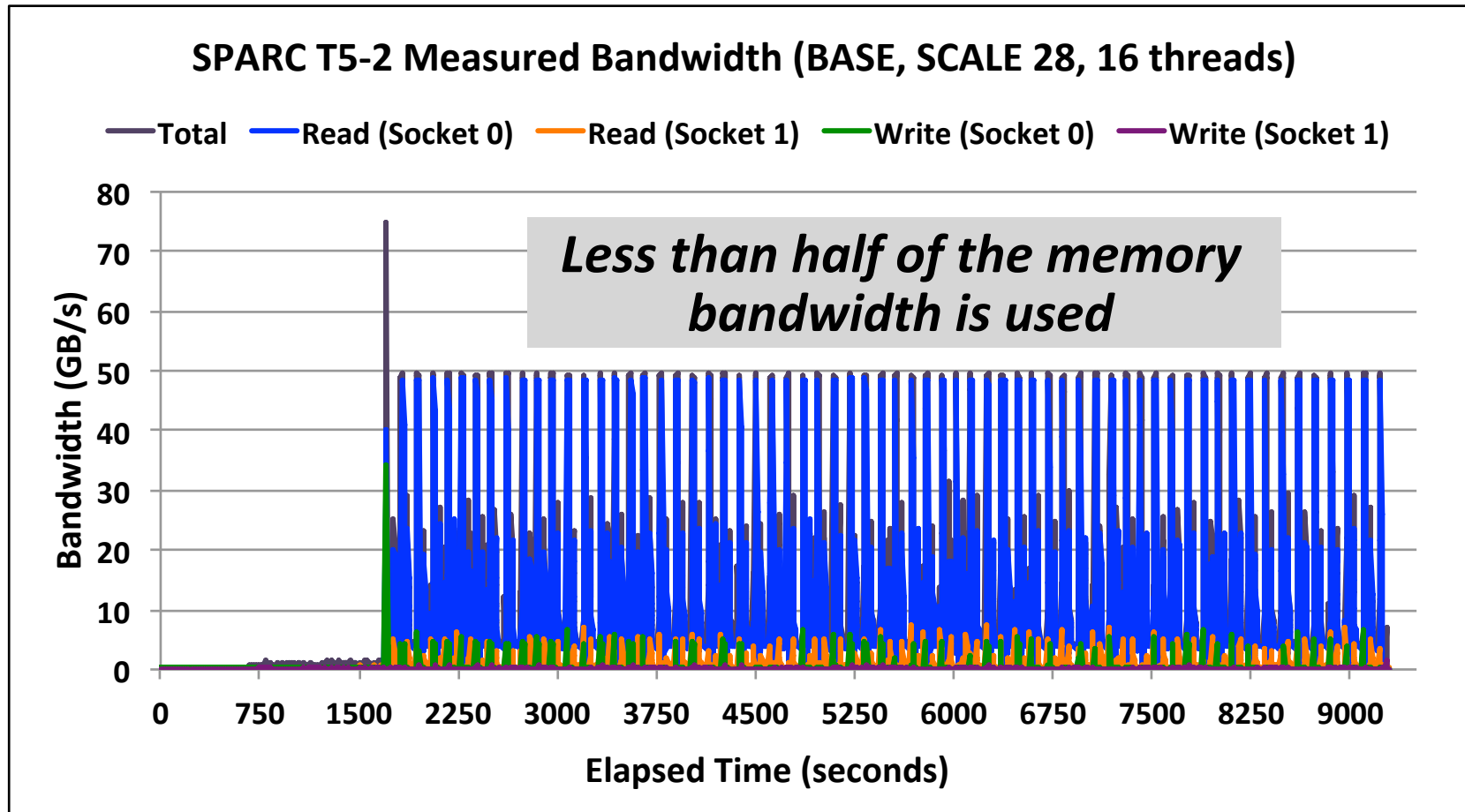
Let's Get Technical



Total CPU Time Distribution



Bandwidth Of The Original Code OpenMP



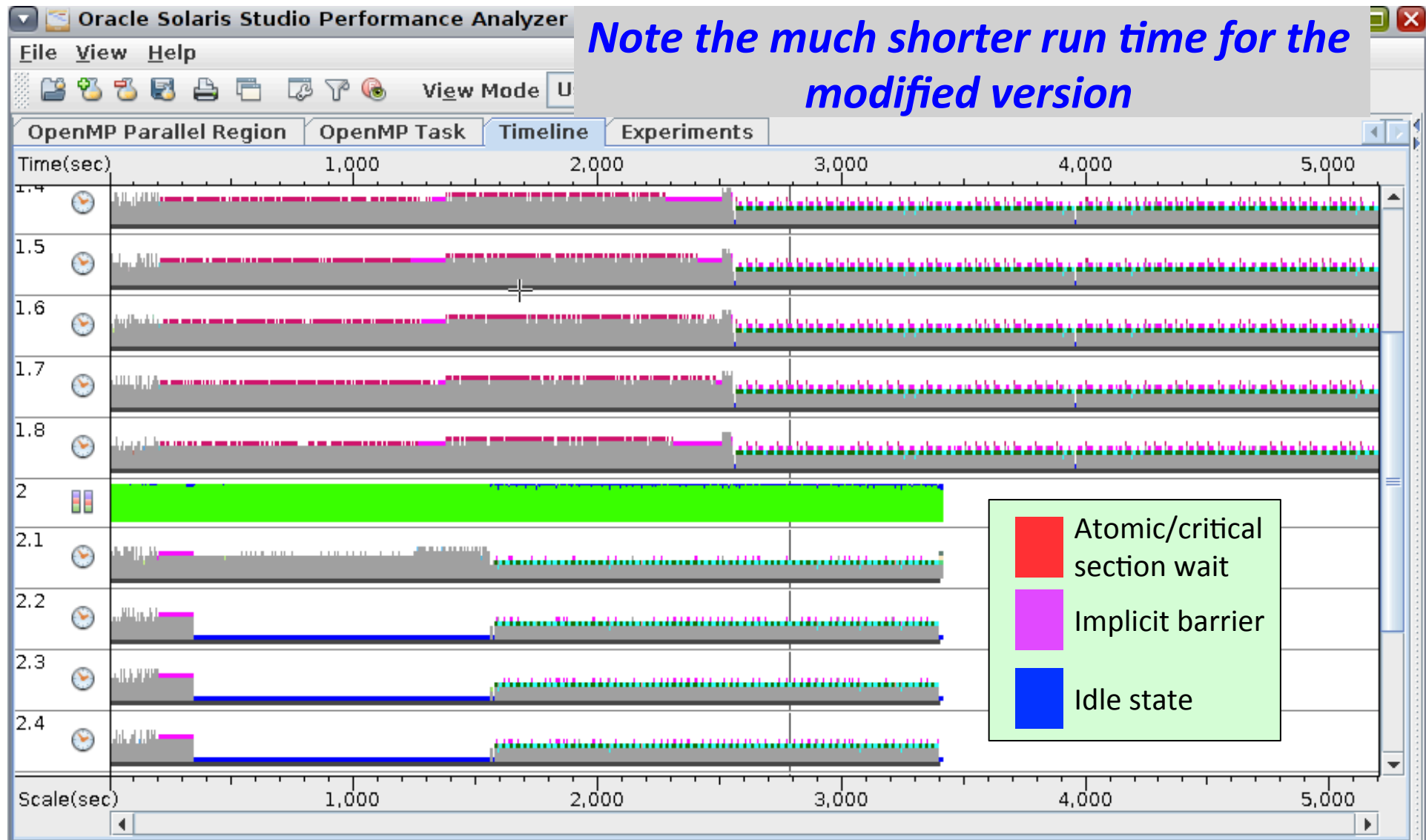
Summary Original Version



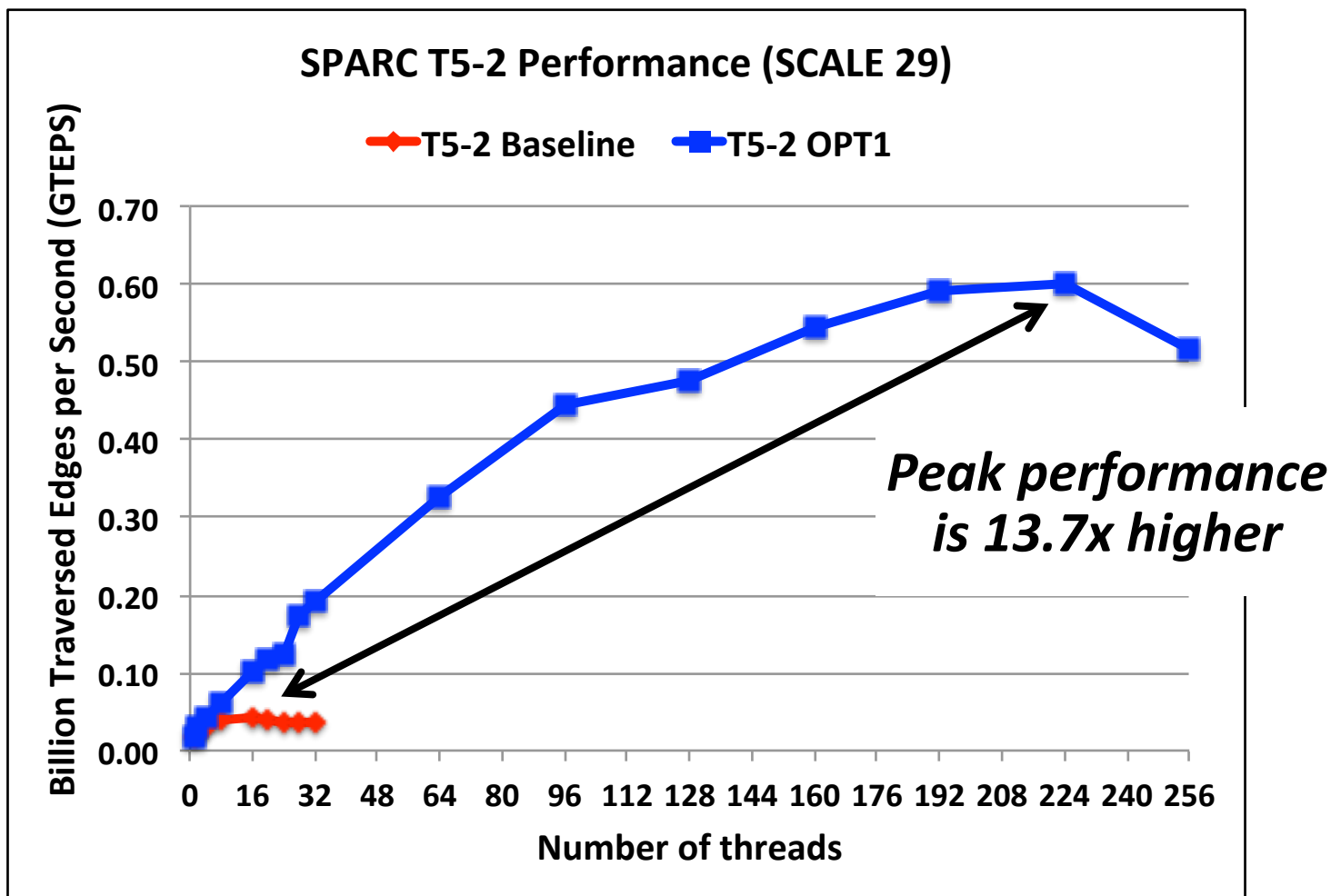
- Communication costs are too high
 - Increases as threads are added
 - This seriously limits the number of threads used
 - This in turn affects memory access on larger graphs
- The bandwidth is not balanced
- Fixes:
 - Find and fix many OpenMP inefficiencies
 - Use some efficient atomic functions



Comparison Of The Two Versions



Performance Comparison





Observations



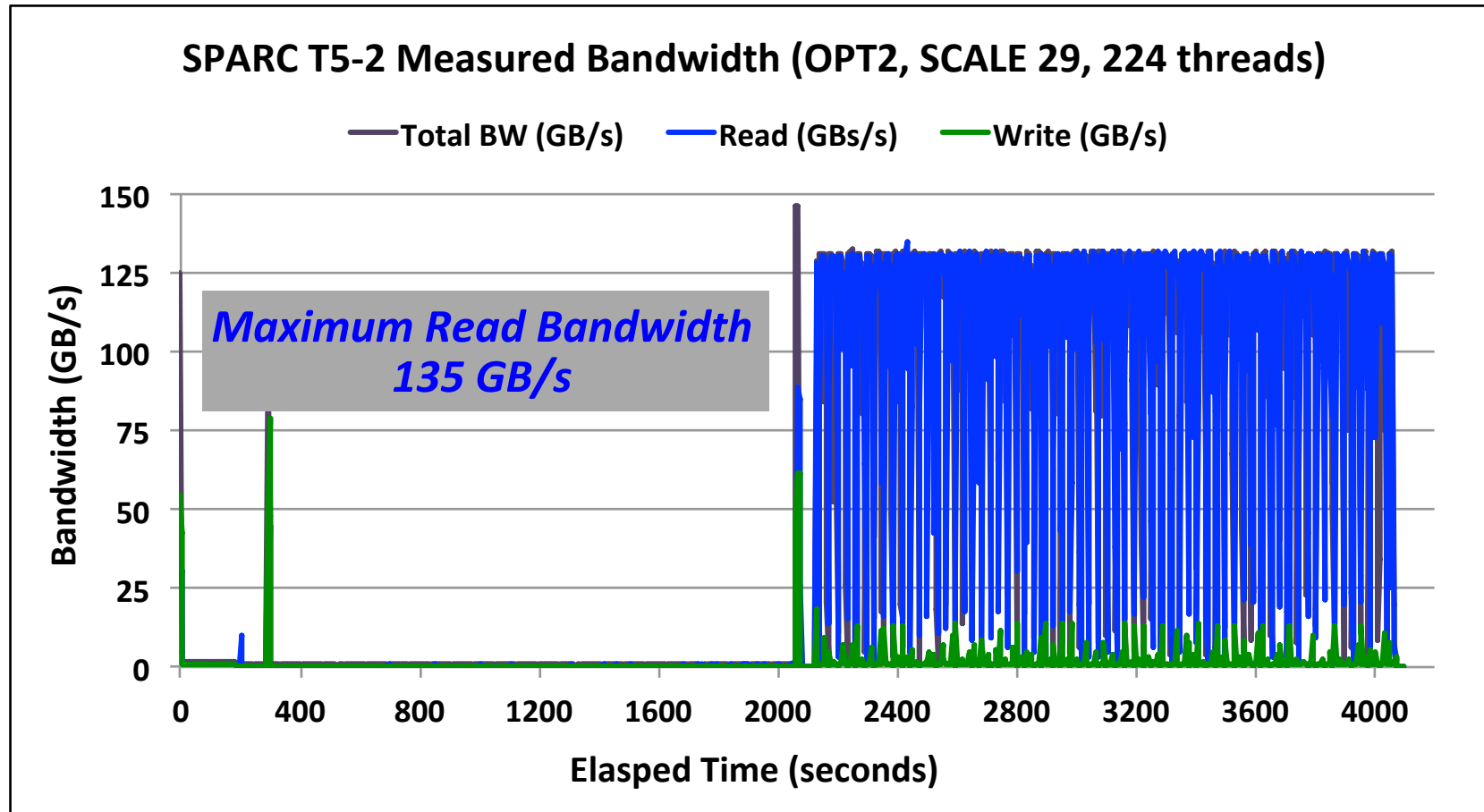
First Touch Placement Is Not Used

The Code Does Not Exploit Large Pages

But Needs It

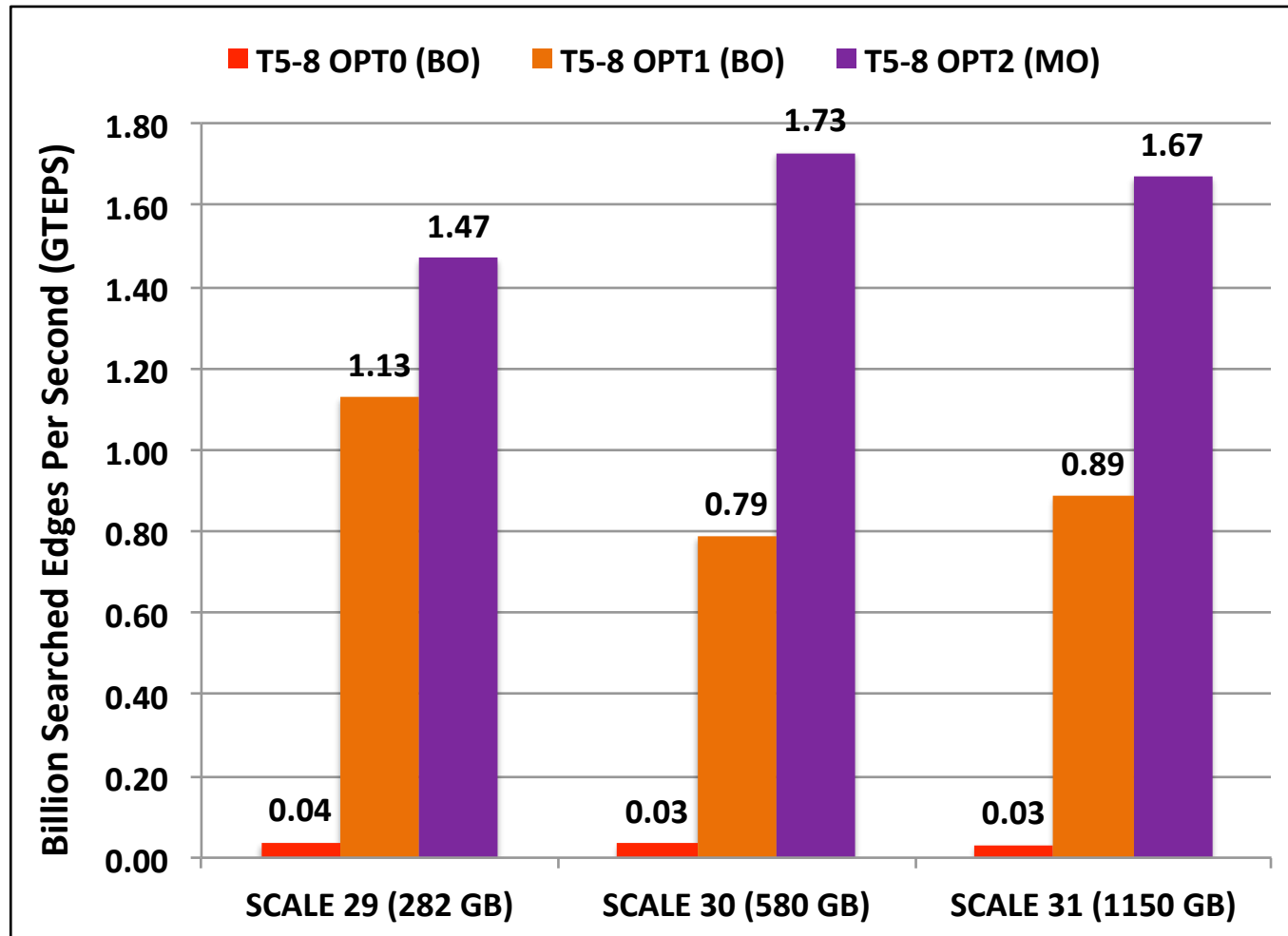
Used A Smarter Memory Allocator

Bandwidth Of The New Code



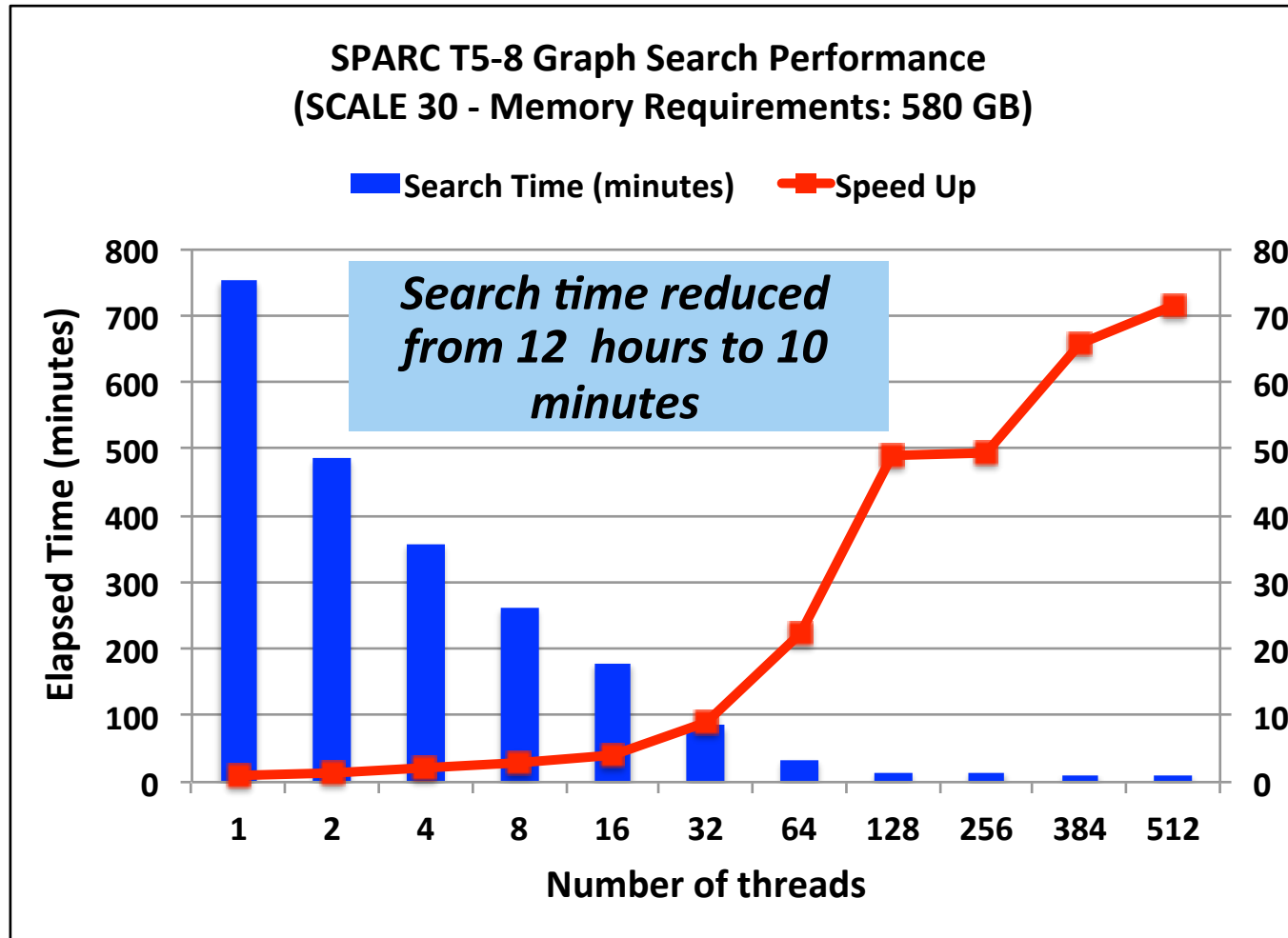
The Result

**39-52x improvement
over original code**



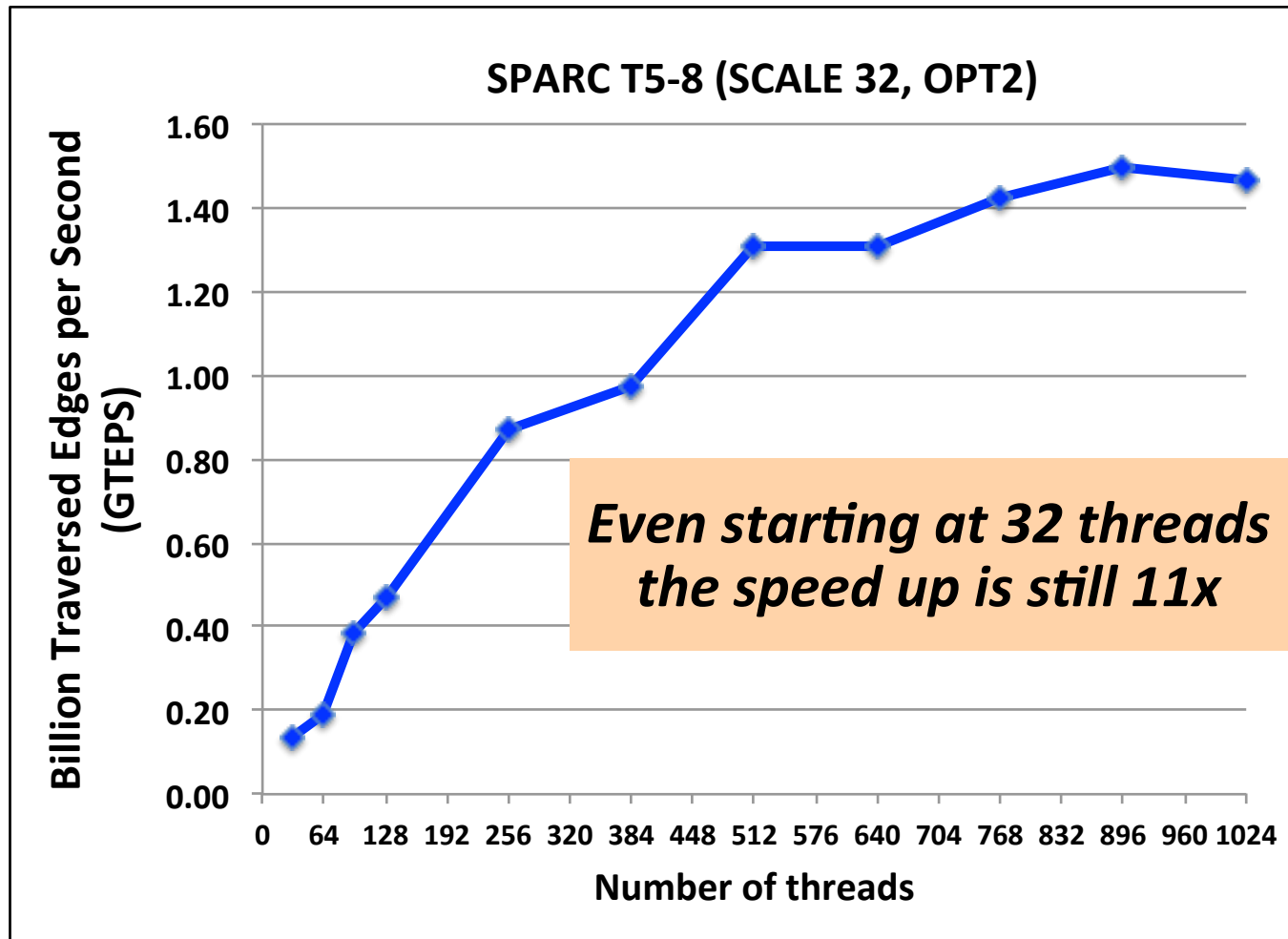
Bigger Is Definitely Better!

**72x
speed up !**

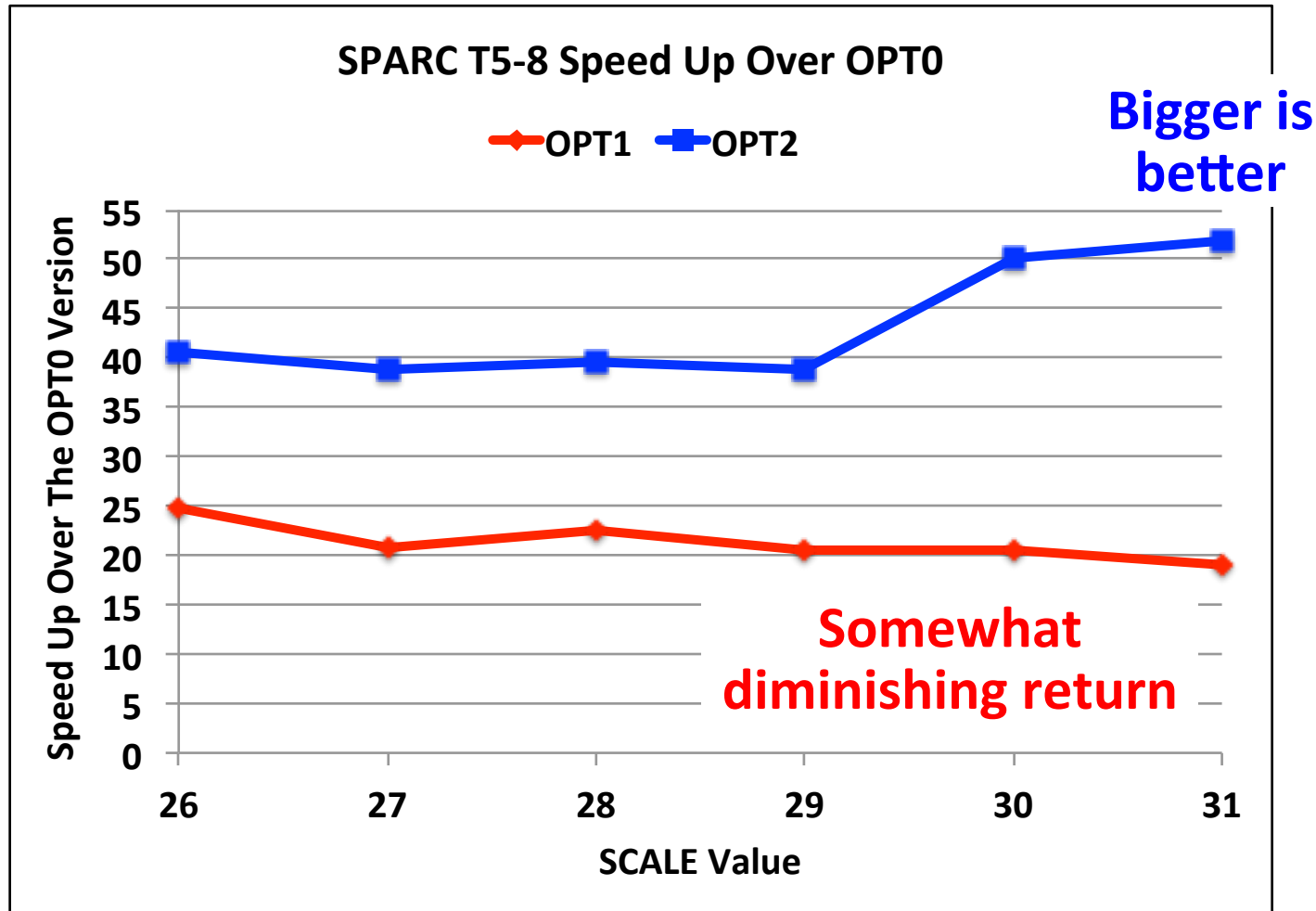


A 2.3 TB Sized Problem

896 Threads !



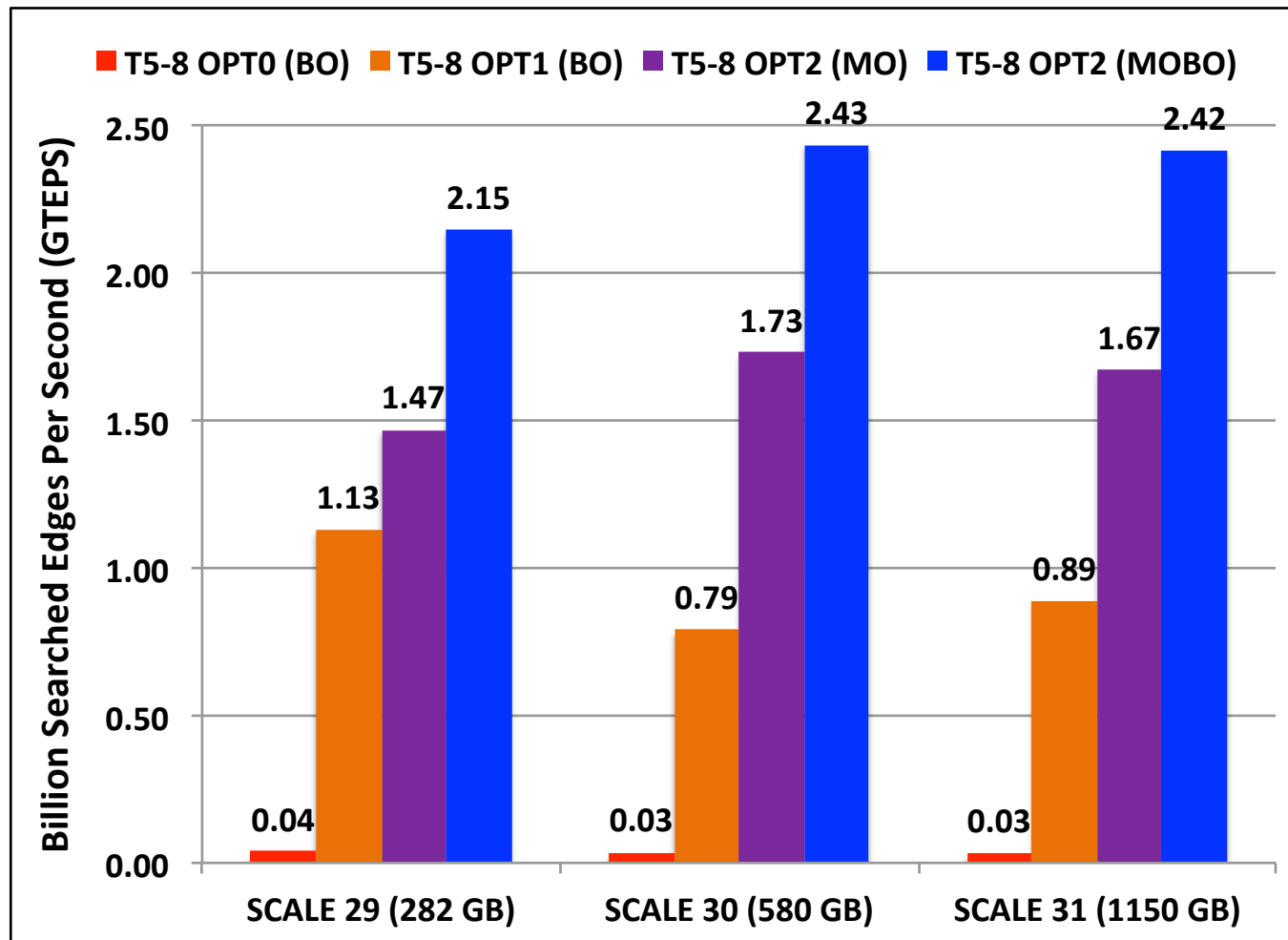
Tuning Benefit Breakdown





A Simple OpenMP Change

**57-75x
improvement**



Methodology



If The Code Does Not Scale Well

Use A Profiling Tool

Use The Checklist To Identify Bottlenecks

Tackle Them One By One

This Is An Incremental Approach

But Very Rewarding

Summary



OpenMP Can Not Not Scale

In Most Cases The Application Has Bottlenecks

Very Often These Can Be Eliminated Or Reduced

For Ultimate Scalability Find And Fix Everything

And Enjoy A Very Scalable Application 😊

Thank You And Stay Tuned !

ruud.vanderpas@oracle.com

The Oracle logo, consisting of the word "ORACLE" in white sans-serif capital letters on a red rectangular background.

ORACLE

A graphic featuring a light green diagonal banner with the text "Bad OpenMP Does Not Scale" in blue, overlaid on a black, pixelated, cloud-like shape.

**Bad OpenMP
Does Not Scale**