Getting OpenMP Up To Speed

Ruud van der Pas
Distinguished Engineer
SPARC Microelectronics
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

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

<table>
<thead>
<tr>
<th><strong>Don’t Use OpenMP</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start With A Serial Code That Performs Badly</strong></td>
</tr>
<tr>
<td><strong>Don’t Use A Profiling Tool</strong></td>
</tr>
<tr>
<td><strong>Excessive Use Of Parallel Regions</strong></td>
</tr>
<tr>
<td><strong>Excessive Use Of Shared Data</strong></td>
</tr>
</tbody>
</table>
### Top 10 Of Dumb Things To Do

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Don’t Use The Nowait Clause</strong></td>
</tr>
<tr>
<td><strong>Use Locks Where They Don’t Make Sense</strong></td>
</tr>
<tr>
<td><strong>Ignore Load Balancing Issues</strong></td>
</tr>
<tr>
<td><strong>Ignore cc-NUMA</strong></td>
</tr>
<tr>
<td><strong>Forget All Of The Above</strong></td>
</tr>
</tbody>
</table>
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 ?”
A key operation on a graph is to search for connections.

The performance is measured by the number 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 $2^{\text{scale}}$
- The number of edges is 16 per vertex
The Initial Performance (35 GB)

SPARC T5-2 Performance (SCALE 26)

Game over beyond 16 threads
That doesn’t scale very well

Let’s use a bigger machine!
Initial Performance (35 GB)

SPARC T5-2 and T5-8 Performance

- T5-2 SCALE 26
- T5-8 SCALE 26

Billion Traversed Edges per Second (GTEPS)

Number of threads

0.000 0.010 0.020 0.030 0.040 0.050
0 4 8 12 16 20 24 28 32 36 40 44 48
Oops! That can’t be true

Let’s run a larger graph!
Initial Performance (280 GB)

SPARC T5-2 and T5-8 Performance

Billion Traversed Edges per Second (GTEPS)

Number of threads

- T5-2 SCALE 29
- T5-8 SCALE 29
Let’s Get Technical
Total CPU Time Distribution

Total CPU Time Percentage Distribution (Baseline, SCALE 26)

- Atomic operations
- OMP-atomic_wait
- OMP-critical_section_wait
- OMP-implicit_barrier
- Other

Number of threads

Function 1
Function 2
Bandwidth Of The Original Code

SPARC T5-2 Measured Bandwidth (BASE, SCALE 28, 16 threads)

- Total
- Read (Socket 0)
- Read (Socket 1)
- Write (Socket 0)
- Write (Socket 1)

Less than half of the memory bandwidth is used
Summary Original Version

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

Secret Sauce

BO
Comparison Of The Two Versions

Note the much shorter run time for the modified version
Performance Comparison

SPARC T5-2 Performance (SCALE 29)

Peak performance is 13.7x higher
BO More Secret Sauce MO
## Observations

<table>
<thead>
<tr>
<th>First Touch Placement Is Not Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Code Does Not Exploit Large Pages</td>
</tr>
<tr>
<td>But Needs It ....</td>
</tr>
<tr>
<td>Used A Smarter Memory Allocator</td>
</tr>
</tbody>
</table>
Bandwidth Of The New Code

SPARC T5-2 Measured Bandwidth (OPT2, SCALE 29, 224 threads)

- Total BW (GB/s)
- Read (GBs/s)
- Write (GB/s)

Maximum Read Bandwidth 135 GB/s
The Result

39-52x improvement over original code

![Bar Chart]

- T5-8 OPT0 (BO)
- T5-8 OPT1 (BO)
- T5-8 OPT2 (MO)

<table>
<thead>
<tr>
<th>Scale</th>
<th>Billion Searched Edges Per Second (GTEPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCALE 29 (282 GB)</td>
<td>1.13</td>
</tr>
<tr>
<td>SCALE 30 (580 GB)</td>
<td>0.79</td>
</tr>
<tr>
<td>SCALE 31 (1150 GB)</td>
<td>0.89</td>
</tr>
</tbody>
</table>

OpenMP Booth – Getting OpenMP Up To Speed
Ruud van der Pas
Bigger Is Definitely Better!

Search time reduced from 12 hours to 10 minutes
A 2.3 TB Sized Problem

896 Threads!

Even starting at 32 threads the speed up is still 11x
Tuning Benefit Breakdown

SPARC T5-8 Speed Up Over OPT0

- OPT1
- OPT2

Bigger is better

Somewhat diminishing return

Tuning Benefit Breakdown

OpenMP Booth – Getting OpenMP Up To Speed
Ruud van der Pas
Different Secret Sauce

MO → MOBO
A Simple OpenMP Change

57-75x improvement

OpenMP Booth – Getting OpenMP Up To Speed
Ruud van der Pas
# Methodology

<table>
<thead>
<tr>
<th>If The Code Does Not Scale Well</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use A Profiling Tool</td>
</tr>
<tr>
<td>Use The Checklist To Identify Bottlenecks</td>
</tr>
<tr>
<td>Tackle Them One By One</td>
</tr>
<tr>
<td>This Is An Incremental Approach</td>
</tr>
<tr>
<td>But Very Rewarding</td>
</tr>
</tbody>
</table>

OpenMP Booth – Getting OpenMP Up To Speed
Ruud van der Pas
### 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