OpenMPCon
USERS CONVENTION

OpenMP TR3 aka future 4.1 Introducing OpenMPCon 2015 OpenMP Code Challenge

Michael Wong

michaelw@ca.ibm.com

OpenMP CEO

Canada and IBM C++ Standard HoD

Vice chair of Standards Council of Canada Programming Languages

Chair of WG21 SG5 Transactional Memory

Director and Vice President of ISOCPP.org

http://bit.ly/sc14-eval

SC14

What now?

- Nearly every C, C++ features makes for beautiful, elegant code for developers (Disclaimer: I love C++)
  - Please insert your beautiful code here:
  - Elegance is efficiency, or is it? Or
  - What we lack in beauty, we gain in efficiency; Or do we?
- The C++11 Std is
  - 1353 pages compared to 817 pages in C++03
- The new C++14 Std is
  - 1373 pages (N3937), vs the free n3972
- The new C11 is
  - 701 pages compared to 550 pages in C99
- OpenMP 3.1 is
  - 346 pages and growing
- OpenMP 4.0 is
  - 538 pages

Beautiful and elegant Lambdas

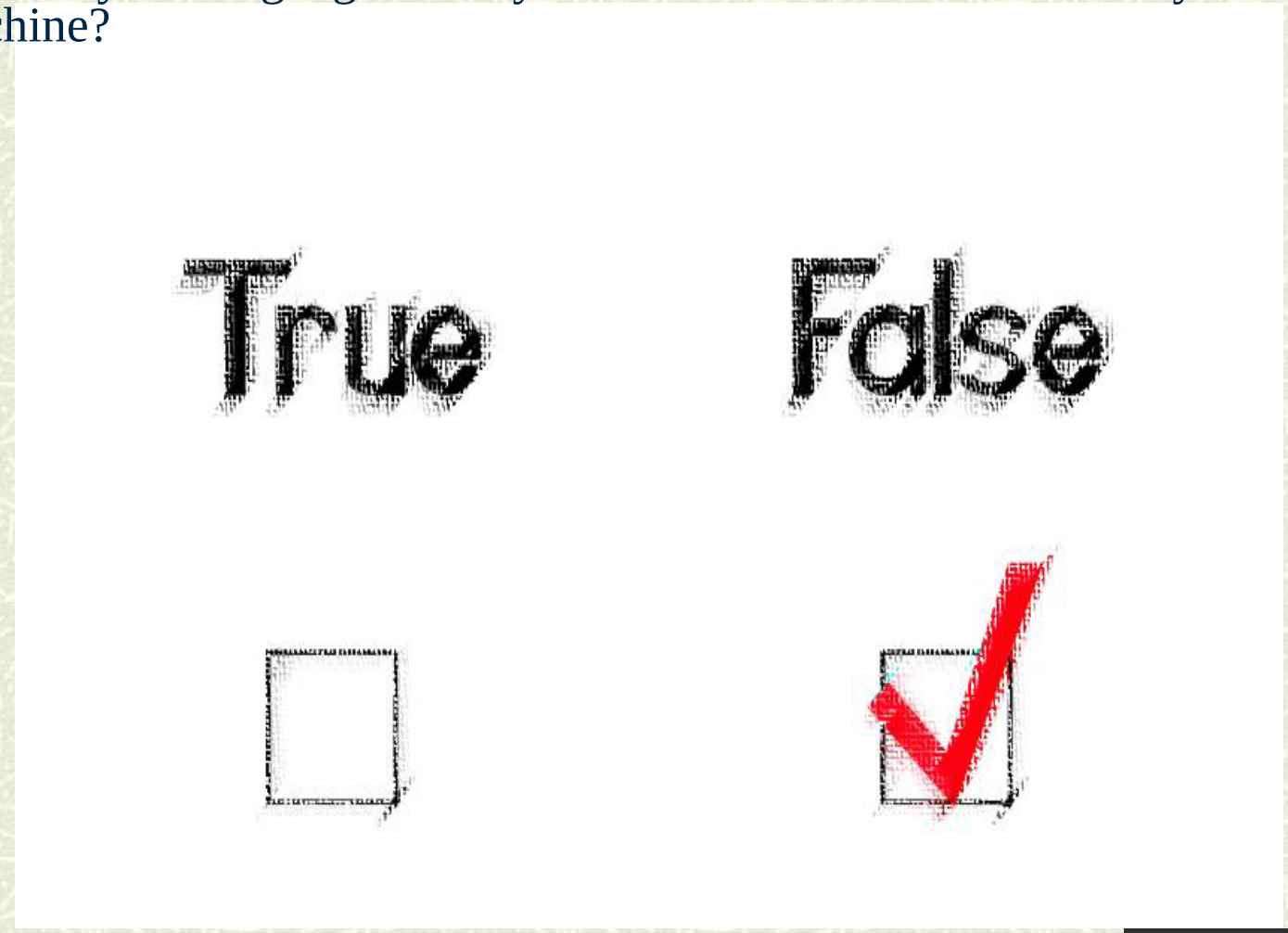
C++98	C++11

vector<int>::iterator i = v.begin();
for(; i != v.end(); ++i) {
  if(*i > x && *i < y)
  break;
}
	auto i = **find_if**(begin(v), end(v),
	[=](int i) {
	  return i > x && i < y;
	});

- "Lambdas, Lambdas Everywhere" *http://vimeo.com/23975522*
- *Full Disclosure: I love C++ and have for many years*
- *But … What is wrong here?*

The Truth

- Q: Does your language allow you to access all the GFLOPS of your machine?



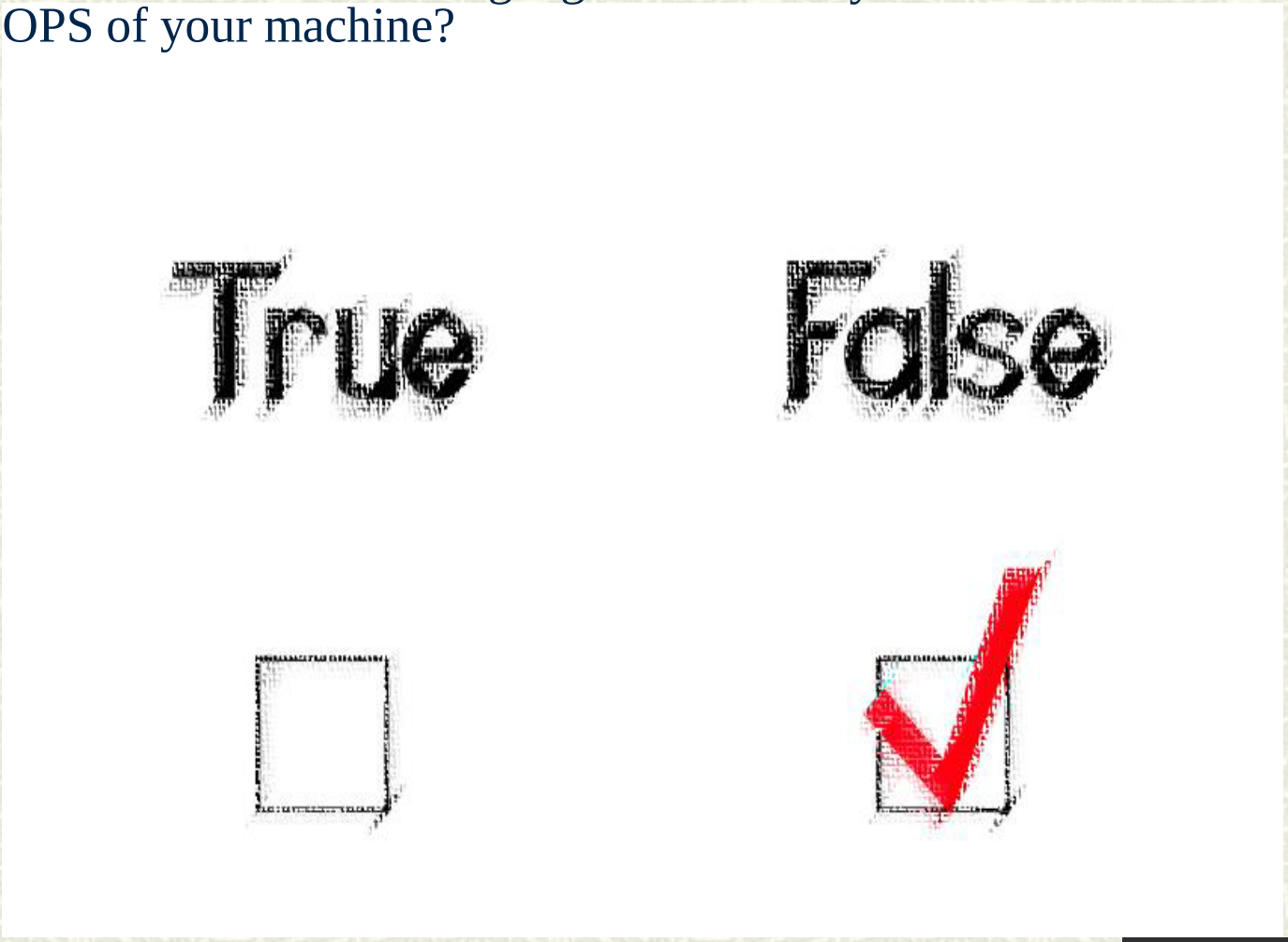
“Is there in Truth No Beauty?”

from *Jordan by George Herbert*

- Q: Does your language allow you to access all the GFLOPS of your machine?
- A: What a quaint concept!
  - I thought its natural to drop out into OpenCL, CUDA, OpenGL, DirectX, C++AMP, Assembler …. to get at my GPU
  - Why? I just use my language as a cool driver, it’s a great scripting language too. But for real kernel computation, I just use Fortran
  - I write vectorized code, so my vendor offers me intrinsics, they also tell me they can auto-vectorize, though I am not sure how much they really do, so I am looking into OpenCL
  - Well, I used to use one thread, but now that I use multiple threads, I can get at it with C++11, OpenMP, TBB, GCD, PPL, MS then continuation, Cilk
  - I know I may have a TM core somewhere, so my vendor offers me intrinsics
  - No I like using a single thread, so I just use C, or C++

The Question

- Q: Is it true that there is a language that allows you to access all the GFLOPS of your machine?



Power of Computing

- 1998, when C++ 98 was released
  - Intel Pentium II: 0.45 GFLOPS
  - No SIMD: SSE came in Pentium III
  - No GPUs: GPU came out a year later
- 2011: when C++11 was released
  - Intel Core-i7: 80 GFLOPS
  - AVX: 8 DP flops/HZ*4 cores *4.4 GHz= 140 GFlops
  - GTX 670: 2500 GFLOPS
- Computers have gotten so much faster, how come software have not?
  - Data structures and algorithms
  - latency

In 1998, a typical machine had the following flops

- .45 GFLOP, 1 core

- Single threaded C++98/C99 dominated this picture

In 2011, a typical machine had the following flops

- 2500 GFLOP GPU

- To program the GPU, you use CUDA, OpenCL, OpenGL, DirectX, Intrinsics, C++AMP

In 2011, a typical machine had the following flops

- 2500 GFLOP GPU+140GFLOP AVX

- To program the GPU, you use CUDA, OpenCL, OpenGL, DirectX, Intrinsics, C++AMP
- To program the vector unit, you use Intrinsics, OpenCL, or auto-vectorization

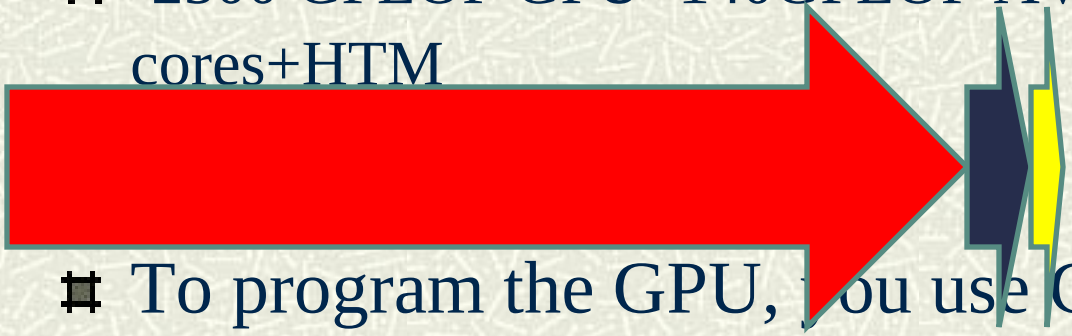
In 2011, a typical machine had the following flops

- 2500 GFLOP GPU+140GFLOP AVX+80GFLOP 4 cores

- To program the GPU, you use CUDA, OpenCL, OpenGL, DirectX, Intrinsics, C++AMP
- To program the vector unit, you use Intrinsics, OpenCL, or auto-vectorization
- To program the CPU, you use C/C++11, OpenMP, TBB, Cilk, MS Async/then continuation, Apple GCD, Google executors

In 2011, a typical machine had the following flops

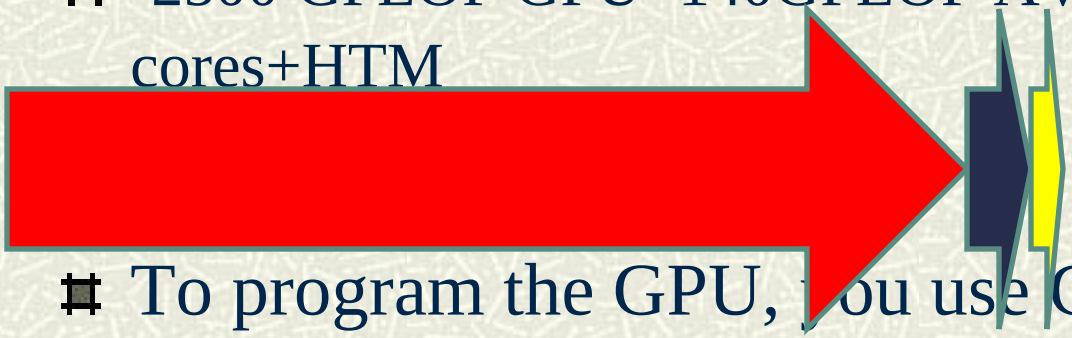
- 2500 GFLOP GPU+140GFLOP AVX+80GFLOP 4 cores+HTM



- To program the GPU, you use CUDA, OpenCL, OpenGL, DirectX, Intrinsics, C++AMP
- To program the vector unit, you use Intrinsics, OpenCL, or auto-vectorization
- To program the CPU, you use C/C++11, OpenMP, TBB, Cilk, MS Async/then continuation, Apple GCD, Google executors
- To program HTM, you have?

In 2014, a typical machine had the following flops

- 2500 GFLOP GPU+140GFLOP AVX+80GFLOP 4 cores+HTM

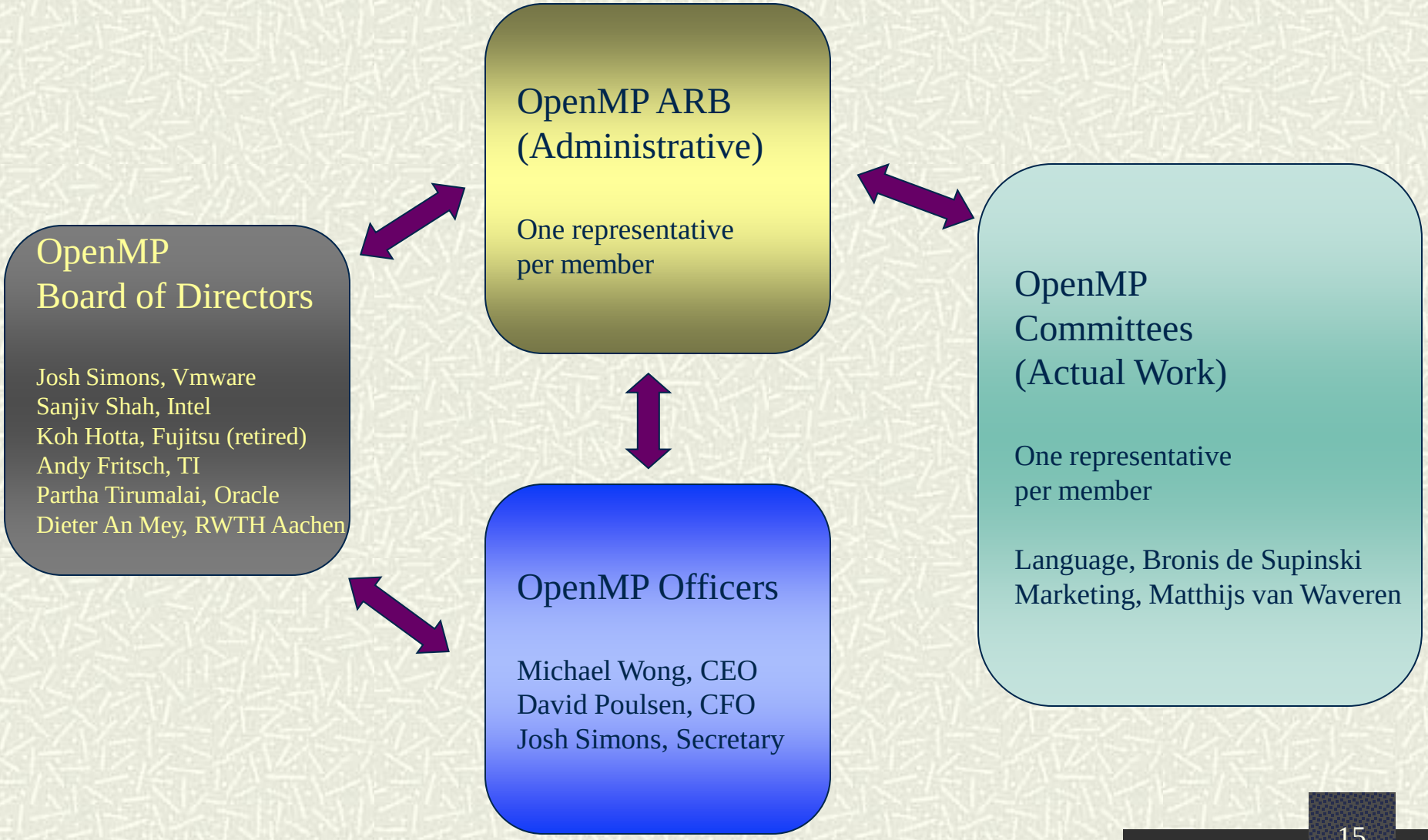


- To program the GPU, you use CUDA, OpenCL, OpenGL, DirectX, Intrinsics, C++AMP, **OpenMP**
- To program the vector unit, you use Intrinsics, OpenCL, or auto-vectorization, **OpenMP**
- To program the CPU, you might use C/C++11/14, **OpenMP**, TBB, Cilk, MS Async/then continuation, Apple GCD, Google executors
- To program HTM, you have? C++/OpenMP TM

Agenda

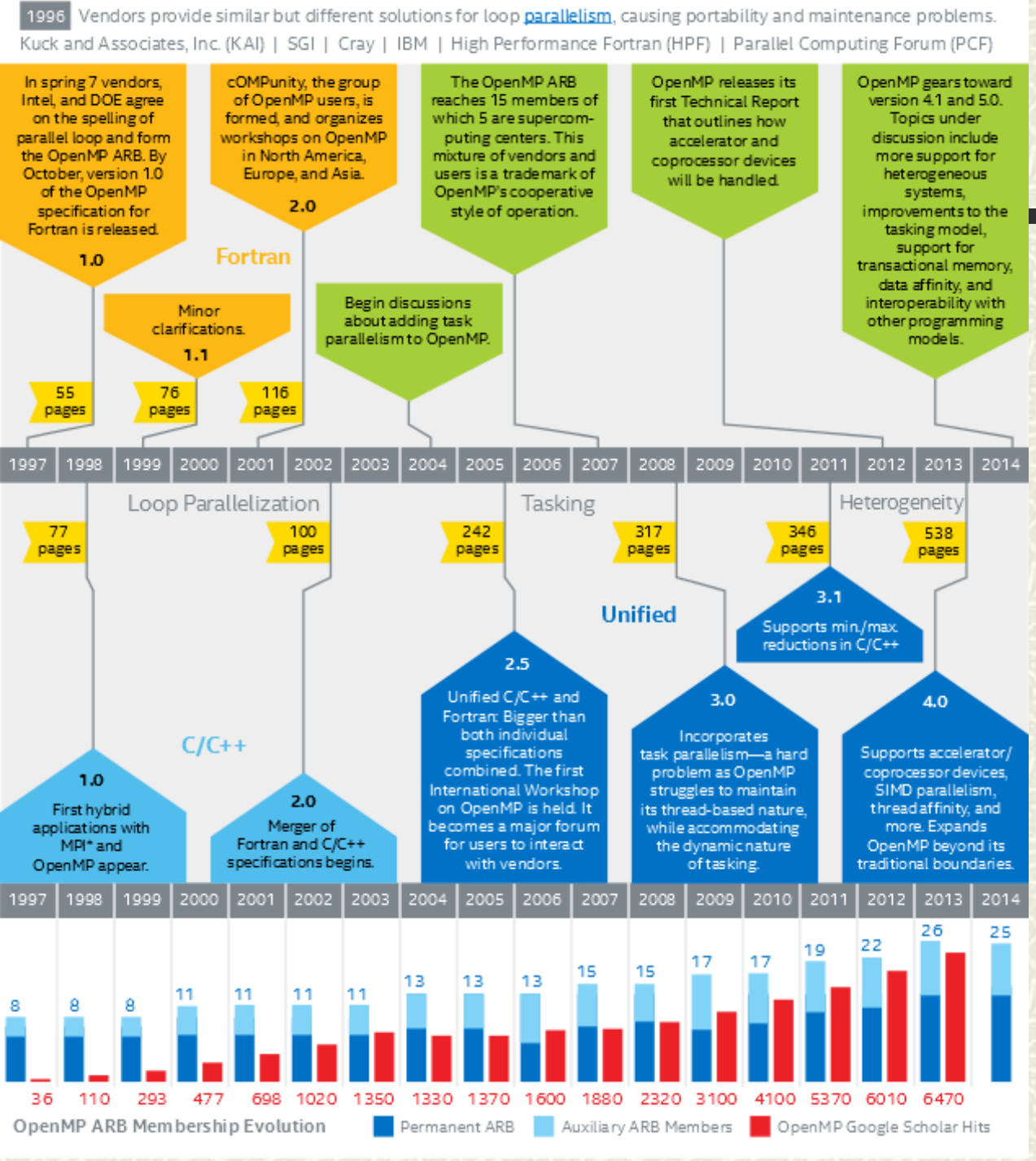
- **The OpenMP ARB**
- History of OpenMP
- OpenMP Code Challenge
- OpenMPCon 2015 and IWOMP2015
- Modernizing OpenMP

OpenMP ARB Current Organization



Agenda

- The OpenMP ARB
- **History of OpenMP**
- OpenMP Code Challenge
- OpenMPCon 2015 and IWOMP2015
- Modernizing OpenMP



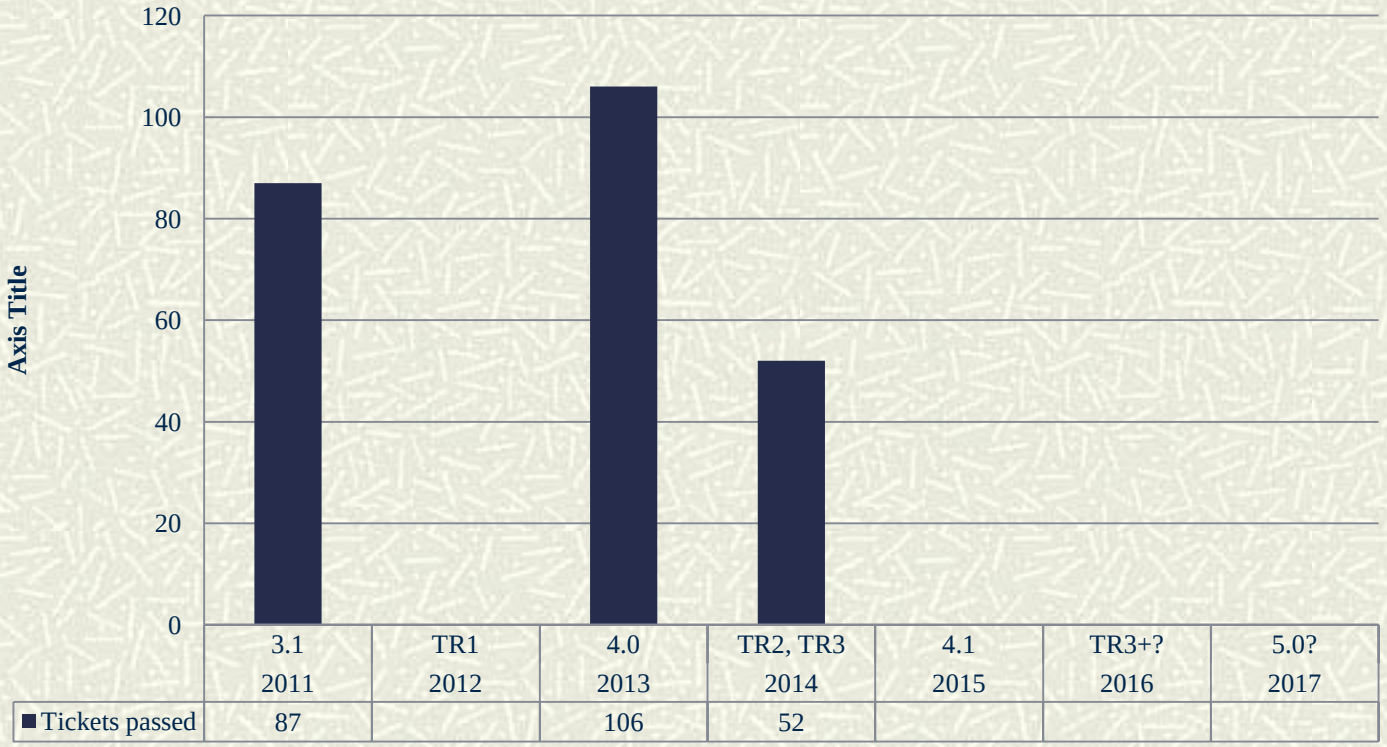
More agile

Next OpenMP revision cycle: **faster**, more **predictable**

Less monolithic: Delivering **concurrent** TRs & language extensions

OpenMP is a living language

Tickets passed



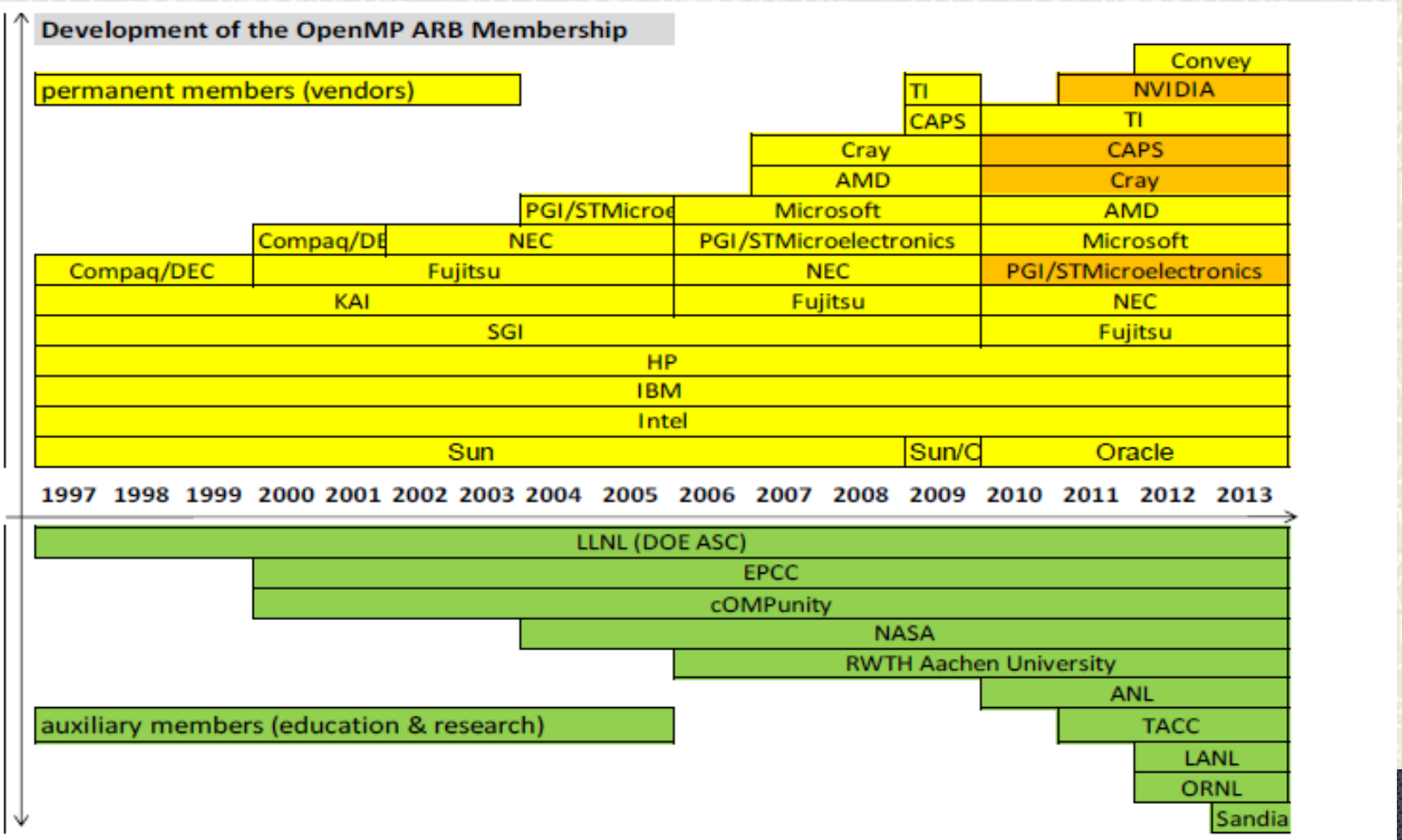
OpenMP internal Organization

Today

OpenMP Members growth

25 members and growing

- From Dieter An Mey, RWTH Aachen 2012
- Added 3 more members since: Redhat/GCC, BSC, UH
- More members coming



Agenda

- The OpenMP ARB
- History of OpenMP
- **OpenMP Code Challenge**
- OpenMPCon 2015 and IWOMP2015
- Modernizing OpenMP

OpenMP is Ready!

We designed OpenMP to be the best, but we want it to be better.
We believe OpenMP is ready for your code, but if you doubt us,

Take our challenge!

Bring us your real application's shared memory or offload code (C++AMP, CUDA, OpenACC, OpenCL, …)
We'll show you how to write it in modern OpenMP.

github.com/OpenMP/Challenge

Challenge

OpenMP Code Challenge

We believe that OpenMP is ready for your code, therefore the OpenMP ARB issued this challenge at SC14:-

> Bring us shared memory or offload code (C++ AMP, CUDA, OpenACC®, OpenCL®, ...) from your **real applications**. We'll show you how to write it in modern OpenMP.

What code do we want?

We're interested in real application code, not just code snippets. We believe that with the release of OpenMP TR3, OpenMP is the best directive based language to use to exploit all of your expensive hardware.

Therefore, the codes we'd like

- Are real codes that are important to you
- Are open source (since we're asking you to push them here, that should be obvious)
- Have build and correctness testing scripts
- Have some documentation so that we can understand at least how to build and test them. Ideally a little more too!

If you are only interested in seeing how we can express specific, small operations, we'll look at those too, but we'd prefer to show you that our specification can handle the complexities of real applications, not just toy kernels.

How do I submit code?

Clone this repository, create a new branch, add your code in a new sub-directory, push back to github in your clone, and then send us a pull request via the "Code Challlenge"list at the OpenMP Forum. We'll pull your code here and start to look at it.

Please include a `licence.txt` (or equivalent file), so that we can be confident that we are allowed to look at your code in an un-encumbered manner. (Without this we won't pull the code).

Agenda

- The OpenMP ARB
- History of OpenMP
- OpenMP Code Challenge
- **OpenMPCon 2015 and IWOMP2015**
- Modernizing OpenMP

OpenMPCon

The Conference for the *Entire* OpenMP Community

Existing OpenMP Conferences

- OpenMP already has a number of successful Global conferences:
  - IWOMP
  - OpenMP Language Meetings
  - SuperComputing BoFs
  - SC Boothtalks/ Tutorials
  - …
- What is wrong with them?

Nothing

Nothing

- These are great conferences that prove their worth through the years.

Nothing

- These are great conferences that prove their worth through the years.
- But each is precisely focused and don't intend to serve the entire OpenMP Community.
- There is no place for OpenMP Users to gather.
- There are islands of OpenMP users.

IWOMP and OpenMPCon

- International Workshop on OpenMP
  - 2014 to be held in Brazil
  - A strongly academic conference, with refereed papers, and a Springer-Verlag published proceeding, few users
    - IWOCL was patterned after IWOMP
- What is missing is a user conference similar to ACCU, pyCON, CPPCON, C++Now
- OpenMPCON
  - A user conference paired with IWOMP
  - Non-refereed, user abstracts
  - 1st one will be held in Europe in 2015 to pair with the 2015 IWOMP

OpenMPCon

- What
  - First annual OpenMPCon
- Who
  - Owned by OpenMP ARB and cOMPunity
  - Managed by a committee of volunteers
- Where
  - RWTH Aachen, Germany
- When
  - September 28, 29, 30 2015
- Why
  - To promote OpenMP and Build the OpenMP Community

Goal of OpenMPCon

- It is our mission to be *the* public forum for face-to-face communication in the OpenMP community for the Universe. We will encourage the best use of OpenMP, but other than that are non-partisan and have no agenda. We will facilitate the community reaching consensus about what constitutes best practice and will support it being taught.

Content

- OpenMPCon will be a community conference
- A Call for Submissions will invite any member of the community to propose a session
- Our volunteer content committees will strive for quality, but work for a diversity of viewpoints and experiences
- Other than “Standard” OpenMP, we have no agenda or partisanship

Content

First year content vision:

- Two days of OpenMPCon paired with IWOMP (2 days) and Language Meeting (4 days)
- 60-100 attendees
- 2 keynotes
- 1 plenary session each day + evening (less technical) sessions
- 2 tracks

Content

- In addition to general conference perhaps:
  - Tutorial Days
  - Community “Unconference”
  - Lightning Talks
  - Coding Contests
  - Posters
  - Job Fair
  - Exhibit Hall/vendor sessions

Management

Short term:

- Run by volunteers
- Helped by OpenMP ARB and cOMPunity
- Focus on fundamental requirements
- No frills
- Create a conference good enough to be improved on

Sponsorships

- Current needs
  - Publicity -
    - Announce to get speakers to submit talks
    - Great talks lead to more attendees
  - Recruitment of speakers
    - In general, speakers are not compensated (other than having their registration waived)
    - We want to attract speakers that may expect to be compensated (at least of the extent of covering their expenses)

Sponsorships

- Early Bird Opportunity

Sponsorship Level	Regular Sponsorship	Early Bird Sponsorship
Platinum	$20,000	$12,000
Gold	$12,000	$7000
Silver	$7000	$3000
Bronze	$3000	$1000

How can you help? Submit a Talk

- How you Use OpenMP
- Guidelines of OpenMP
- Tips, tricks, gotchas of OpenMP
- Performance of OpenMP
- Conformance, build, options of OpenMP compilers/runtime implementations
- Latest OpenMP and older releases and how it changes life for users
- Accelerator, and heterogeneous computing (GPUs, accelerators, FPGA, etc.)
- Tasks, task groups and task dependencies
- Scientific, HPC workloads
- Commercial, analytic workload
- Oil and Gas use of OpenMP
- Research on OpenMP (although a good research paper should be submitted to IWOMP)
- Manufacturing, games, graphics, operating system, device drivers, networking domains for OpenMP
- Comparisons with other parallel languages
- Future of OpenMP
- Reduction
- Base language (C++, C, Fortran) with OpenMP
- Affinity
- SIMD and vectors
- Memory model, atomics, locks, mutexes, barriers and critical sections in OpenMP
- OpenMP APIs and environment variables
- Cancellation and Error Model
- Technical Reports, TR1, TR2
- Arrays shaping in OpenMP
- OpenMP concurrency and parallelism
- OpenMP designs
- Cancellation
- Interoperability
- Event programming, futures
- Governance, ARB, committee logistics
- OpenMP website, discussion groups, forums, twikis, conferences, language meetings, IWOMP
- Speculative execution and transactional memory
- Software development tools for OpenMP
- Relation with other (MPI, OpenCL, OpenACC, ISO, etc)
- History of OpenMP
- Tutorial style sessions

What can you expect at OpenMPCon

- **Invited talks and panels**: An OpenMPCon keynote by a noted speaker will start off the conference, followed by talks and sessions full of insight from some of the world's leading experts in OpenMP. Have OpenMP questions? Ask them at one of the OpenMPCon panels featuring those at the cutting edge of the language.
- **Presentations by the OpenMP community**: What do embedded systems, game development, high frequency trading, and particle accelerators have in common? OpenMP, of course! Expect talks from a broad range of domains focused on practical OpenMP techniques, libraries, and tools.
- **Tips and Tricks in OpenMP**: What language serves three General Purpose languages, but also has high-level support for SIMD/vectors, accelerators. The answer is OpenMP. But using it in the most efficient way requires you to know not only the ins and outs, but also the original spirit and intent of the language design.
- **Lightning talks**: Get informed at a fast pace during special sessions of short, informal talks. Never presented at a conference before? This is your chance to share your thoughts on an OpenMP-related topic in an informal setting.
- **Evening events and "unconference" time**: Relax, socialize, or start an impromptu coding session.

Key Dates

- Call for Submission: 2015-03-25
- Submission deadline: 2015-06-01
- Decision notifications by: 2015-06-13
- Fully scheduled program available: 2015-06-27

- Super Early Bird Deadline: 2015-05-01
- Student Registration: 2015-05-06
- Early Bird Deadline: 2015-07-01
- Call for Volunteers: 2015-07-01
- Program Sessions Online: 2015-07-15
- Program Schedule Online: 2015-07-29
- Registration Deadline: 2015-09-23

- OpenMPCon: 2015-09-28/29/30
- IWOMP: 2015-10-01/02

OpenMP

IT Center | RWTH AACHEN UNIVERSITY

IWOMP 2015 in Aachen

We are pleased to announce that **IWOMP 2015** and the first annual **OpenMPCon** will take place in **Aachen** in September 2015 and will be hosted by the **IT Center** of RWTH Aachen University.

International Workshop on OpenMP (IWOMP) is an annual workshop dedicated to the promotion and advancement of all aspects of parallel programming with OpenMP. It is the premier forum to present and discuss issues, trends, recent research ideas, and results related to parallel programming with OpenMP. The international workshop affords an opportunity for OpenMP users as well as developers to come together for discussions and to share new ideas and information on this topic. **www.iwomp.org**

OpenMPCon is the annual, face-to-face gathering, organized by the OpenMP community for the community. Enjoy a keynote, inspirational talks, and a friendly atmosphere designed to help attendees learn from each other, meet interesting people, and generally have a stimulating experience. Multiple diverse technical tracks are being formulated that will appeal to anyone, from the OpenMP novice to the seasoned expert. **www.openmpcon.org**

Agenda

- The OpenMP ARB
- History of OpenMP
- OpenMP Code Challenge
- OpenMPCon 2015 and IWOMP2015
- **Modernizing OpenMP**

The Future Mission of OpenMP

- OpenMP's new mission statement
  - "Standardize directive-based multi-language high-level parallelism that is performant, productive and portable"
  - Updated from
    - "Standardize and unify shared memory, thread-level parallelism for HPC"
- We are open for business:
  - To be the most capable Parallel programming model

OpenMP spec host on Github

- https://github.com/OpenMP/spec
- Private right now
- Latex enables easier and concurrent updates

Major Website update

- Repost on new server and software
- Prepare for multiple devices
  - any screen size,
  - windows to phones
  - touch-friendly/-enabled
  - designed for offline reading
- Links to Stack Overflow, Reddit on OpenMP related questions

Wikibook publishing

- Multiple chapters writing started on
  - Intro/History
  - Accelerators
  - Vector/SIMD
  - Tasks
  - Error
  - Affinity

Membership/dues realignment

- Old concept of Permanent and Auxiliary is
  - Soon to be GONE
- Talking about new categories with dues increase to remain competitive
  - ARB Board: $6K
  - ARB Contributor: $3K
  - Academic: $99

Compilers coming

- GCC 4.9 have all of OpenMP 4.0 except …
  - Coming in 5.0: full target support
- Intel compiler supports Accelerators
- Clang support for OpenMP syntax support completed and released in Clang 3.5
  - Coming in 3.6: full code gen + initial accelerator
  - Coming after that: Performance improvements
- Cray, TI, Oracle, Fujitsu also have products

Future meetings

- http://openmp.org/calendar.html
- Winter 2015: Feb 2-6, Nvidia Santa Clara
- Spring 2015: May 18-22, BSC, Barcelona, Spain
- Fall 2015: RWTH Aachen, Germany
  - Sept 28-30: OpenMPCON
  - Sept 30: Tutorial
  - Oct 1-2: IWOMP 2015
  - Oct 4-7: Language Meeting
- Continue to drive to ratify 4.1 and 5.0

OpenMP future

- **More agile**
- More rapid releases
- More technical Reports
- More consumer-style parallelism
- Deliver faster then ISO
- Deliver experimental and proprietary parallelism
- Allows you to be productive on several languages
- Supported by many vendors

OpenMP internal Organization

Today	Accel	Error	Task	Tools	Affinity	Fortran 2003
Future	TM	Async/Event	Interop	C++11	C11	Memory Model, Loops, Object oriented

Technical Report

This technical report describes possible future directions or extensions to the OpenMP Specification.

The goal of this technical report is to build more widespread existing practice for an expanded OpenMP. It gives advice on extensions or future directions to those vendors who wish to provide them possibly for trial implementation, allows OpenMP to gather early feedback, support timing and scheduling differences between official OpenMP releases, and offers a preview to users of the future directions of OpenMP with the provision stated in the next paragraph.

This technical report is non-normative. Some of the components in this technical report may be considered for standardization in a future version of OpenMP, but they are not currently part of any OpenMP Specification. Some of the components in this technical report may never be standardized, others may be standardized in a substantially changed form, or it may be standardized as is in its entirety.

OMPT: An OpenMP Tools Application Programming Interface for Performance Analysis

OMPT Design Objectives

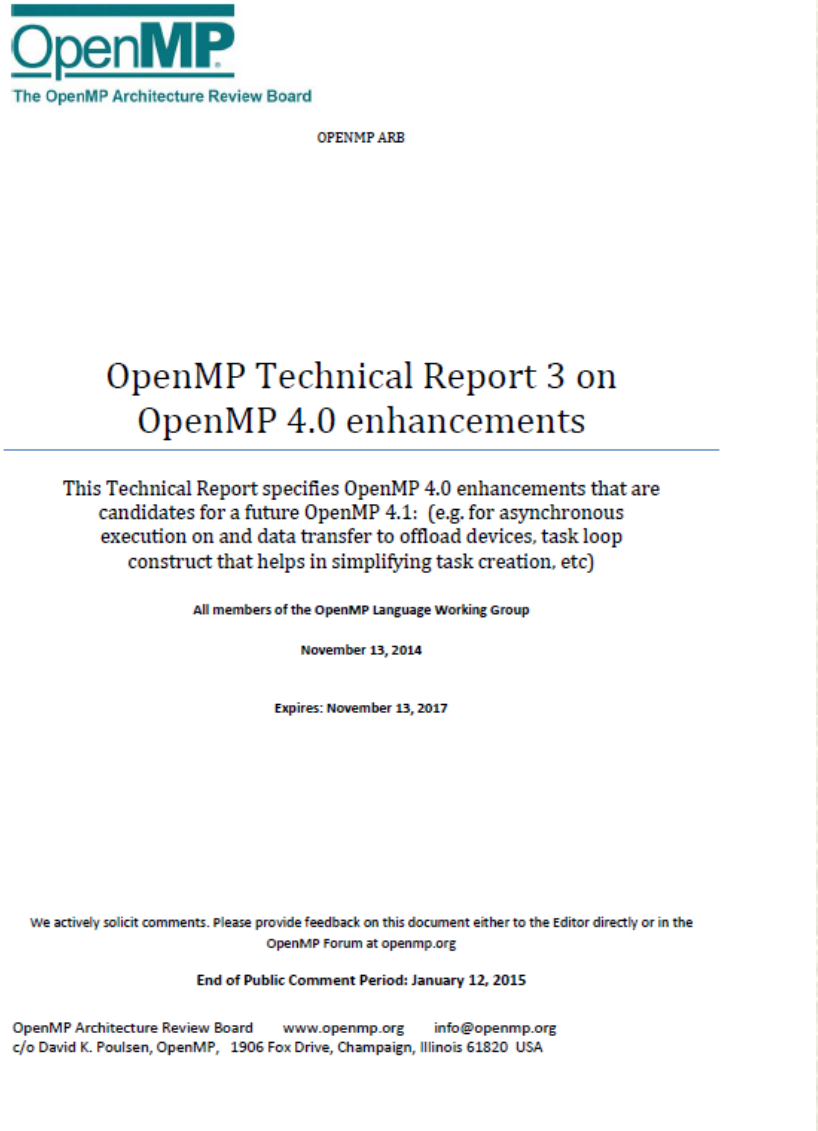
- Enable tools to gather information and associate costs with application source and runtime system
  - provide an interface sufficient to construct low-overhead performance tools based on asynchronous sampling
  - enable a profiler that uses call stack unwinding to identify which frames in its callstack correspond to routines in OpenMP runtime
  - associate activity of a thread at any point in time with a state
    - enable performance tools to monitor behavior
- Negligible overhead if OMPT interface is not in use
- Define support for trace-based performance tools
- Don't impose an unreasonable development burden on
  - runtime implementers
  - tool developers

OMPT Performance Tools API

Overview and Goals

- Create a standardized performance tool interface for OpenMP
  - prerequisite for portable performance tools
  - goal: inclusion in the OpenMP standard
  - role model: PMPI and MPI_T
- Focus on minimal set of functionality
  - provide essential support for sampling-based tools
  - only require support for tools attached at link-time or program launch
- Minimize runtime cost
  - reduce cost in runtime and tool where possible
  - enable integration into optimized runtimes
  - make support for higher-overhead features optional
    - callbacks for blame shifting
    - callbacks for full-featured tracing tools

TR3 aka OpenMP 4.1 features



OpenMP
The OpenMP Architecture Review Board

OPENMP ARB

OpenMP Technical Report 3 on
OpenMP 4.0 enhancements

This Technical Report specifies OpenMP 4.0 enhancements that are candidates for a future OpenMP 4.1: (e.g. for asynchronous execution on and data transfer to offload devices, task loop construct that helps in simplifying task creation, etc)

All members of the OpenMP Language Working Group

November 13, 2014

Expires: November 13, 2017

We actively solicit comments. Please provide feedback on this document either to the Editor directly or in the OpenMP Forum at openmp.org

End of Public Comment Period: January 12, 2015

OpenMP Architecture Review Board www.openmp.org info@openmp.org
c/o David K. Poulsen, OpenMP, 1906 Fox Drive, Champaign, Illinois 61820 USA

- Slides courtesy of Bronis

TR3: An initial 4.1 comment draft

- **refines OpenMP device constructs significantly**
  - Added flush to several device constructs
  - Unstructured data movement
  - Require assignment for map (always)
  - Improved asynchronous execution
    - Could have been in a task with just a target region
    - target and other device regions are now tasks
      - By default, undeferred
      - Can use nowait and depend clauses
  - Many clarifications and minor corrections

Unstructured Data Movement

- Recast device data environment semantics
  - An implicit `target` region surrounds each initial thread
  - That region has ICVs are manipulated in the same ways as host ICVs
  - Variables are mapped into and out of that region's environment
- Use `target enter data` directive to map variables to devices

#pragma omp target enter data [clause [[,] clause] …]

  - Only `to` and `alloc` map types are allowed
- Use `target exit data` directive to unmap variables

#pragma omp target exit data [clause [[,] clause] …]

  - Only `from`, `release` and `delete` map types are allowed
- Semantics of `map` clauses recast as manipulating reference counts
- Use `map` clauses on `target` and `target data` directives to combine `enter` and `exit` semantics

Handle dependence within loops

- Added parameter to `ordered` clause
  - Specifies number of loops covered by `ordered` constructs
  - Similar to `collapse` clause
- Added clauses to ordered directive

#pragma omp ordered [clause [[,] clause] …]

  - Use `threads` with a structured data block for previous (default) semantics
  - Use `simd` with a structured data block to restrict to a single SIMD lane
  - Use `depend` without a structured data block for doacross semantics
- Doacross semantics restrict (but not prohibit) parallelism
  - Automates transformations such as loop skewing
  - Requires new dependence types:
    - `source` specifies iteration completes cross-iteration dependences
    - `sink` with `vec` parameter blocks execution until specified source iterations comp

Specify iterations of a loop should be divided into tasks

- Use `taskloop` directive to execute loop(s) as tasks

#pragma omp taskloop [clause [[,] clause] …]

  - Allows threads in team not to participate in loop execution
  - Supports concurrent asynchronous operations in non-loop tasks

- Partitions iterations of following loop into tasks
  - Avoids manual loop fission to aggregate iterations
  - Can use `grainsize` clause to specify iterations per task
    - Specifies a range of of iterations to support efficient task generation
  - Can use `num_tasks` clause to specify an exact number of tasks
- Accepts most clauses applicable to loop and task constructs
  - `if` generates undeferred tasks (i.e., it has tasking semantics)
  - `reduction` support not yet provided (may wait until OpenMP 5.0)

Other TR3 refinements

- Many clarifications and minor enhancements
  - SIMD extensions
  - Thread affinity policies
  - Various other locations
- Reductions for C/C++ arrays
- Improved support for C++ reference types
- Reduced list of unsupported Fortran 2003 features by four items
- New terms to simplify discussion of new features

Feedback

- http://bit.ly/sc14-eval