OpenMP 4.0: A Significant Paradigm Shift in Parallelism

Michael Wong
OpenMP CEO
michaelw@ca.ibm.com
OpenMP 4.0 Specifications Released

The OpenMP 4.0 API Specification is released with Significant New Standard Features

The OpenMP 4.0 API supports the programming of accelerators, SIMD programming, and better optimization using thread affinity.

The OpenMP Consortium has released OpenMP API 4.0, a major upgrade of the OpenMP API standard language specifications. Besides several major enhancements, this release provides a new mechanism to describe regions of code where data and/or computation should be moved to another computing device.

Bronis R. de Supinski, Chair of the OpenMP Language Committee, stated that "OpenMP 4.0 API is a major advance that adds two new forms of parallelism in the form of device constructs and SIMD constructs. It also includes several significant extensions for the loop-based and task-based forms of parallelism already supported in the OpenMP 3.1 API."

The 4.0 specification is now available on the OpenMP Specifications page.

Standard for parallel programming extends its reach

With this release, the OpenMP API specifications, the de-facto standard for parallel programming on shared memory systems, continues to extend its reach beyond pure HPC to include DSPs, real time systems, and accelerators. The OpenMP API aims to provide high-level parallel language support for a wide range of applications, from automotive and aeronautics to biotech, automation, robotics and financial analysis.

New features in the OpenMP 4.0 API include:

- **Support for accelerators.** The OpenMP 4.0 API specification effort included significant participation by all the major vendors in order to support a wide variety of compute devices. OpenMP API provides mechanisms to describe regions of code where data and/or computation should be moved to another computing device. Several prototypes for the accelerator proposal have already been implemented.

- **SIMD constructs to vectorize both serial as well as parallelized loops.** With the advent of SIMD units in all major processor chips, portable support for accessing them is essential. OpenMP 4.0 API provides mechanisms to describe when multiple iterations of the loop can be executed concurrently using SIMD instructions and to describe how to create versions of functions that can be invoked across SIMD lanes.
Agenda

- The OpenMP ARB
- History of OpenMP
- OpenMP 4.0 Parallelism
- OpenMP Future Parallelism
- OpenMP new Mission Statement
OpenMP ARB Current Organization

OpenMP ARB (Administrative)
One representative per member

OpenMP Board of Directors
Josh Simons, VMware
Sanjiv Shah, Intel
Koh Hotta, Fujitsu
Andy Fritsch, TI
Partha Tirumalai, Oracle

OpenMP Officers
Michael Wong, CEO
David Poulsen, CFO
Josh Simons, Secretary

OpenMP Committees (Actual Work)
One representative per member
Language, Bronis de Supinski
Marketing, Matthijs van Waveren
Tools, Martin Schultz,
Robert Dietrich
Agenda

- The OpenMP ARB
- History of OpenMP
- OpenMP 4.0 Parallelism
- OpenMP Future Parallelism
- OpenMP new Mission Statement
A brief history of OpenMP API

1997
- Fortran V1.0
1998
- Fortran V1.1
- C & C++ V1.0
2001
- Fortran V2.0
- C & C++ V2.0
2002
- Fortran, C & C++ V2.5
2004
- Fortran, C & C++ V3.0
2005
- Fortran, C & C++ V3.1
2010
- Fortran, C & C++ V4.0

2014 onwards, more agile

Next OpenMP revision cycle: faster, more predictable
Less monolithic: Delivering concurrent TRs & language extensions
OpenMP is a living language
OpenMP internal Organization

OpenMP ARB

Language WG

Marketing WG

Today

Accel  Error  Task  Tools  Affinity  Fortran 2003
OpenMP Members growth

From Dieter An Mey, RWTH Aachen 2012

Development of the OpenMP ARB Membership

permanent members (vendors)
- Compaq/DEC
- Fujitsu
- KAI
- SGI
- HP
- IBM
- Intel
- Sun
- Sun/C
- Oracle

auxiliary members (education & research)
- LLNL (DOE ASC)
- EPCC
- cOMPunity
- NASA
- RWTH Aachen University
- ANL
- TACC
- LANL
- ORNL
- Sandia

26 members and growing
Members added since 2012

- Red Hat/GCC
- Barcelona SuperComputing Centre
- University of Houston
Agenda

- The OpenMP ARB
- History of OpenMP
- OpenMP 4.0 Parallelism
- OpenMP Future Parallelism
- OpenMP new Mission Statement
OpenMP 4.0 Specifications Released

The OpenMP 4.0 API Specification is released with Significant New Standard Features

The OpenMP 4.0 API supports the programming of accelerators, SIMD programming, and better optimization using thread affinity.

The OpenMP Consortium has released OpenMP API 4.0, a major upgrade of the OpenMP API standard language specifications. Besides several major enhancements, this release provides a new mechanism to describe regions of code where data and/or computation should be moved to another computing device.

Bronis R. de Supinski, Chair of the OpenMP Language Committee, stated that “OpenMP 4.0 API is a major advance that adds two new forms of parallelism in the form of device constructs and SIMD constructs. It also includes several significant extensions for the loop-based and task-based forms of parallelism already supported in the OpenMP 3.1 API.”

The 4.0 specification is now available on the OpenMP Specifications page.

Standard for parallel programming extends its reach

With this release, the OpenMP API specifications, the de-facto standard for parallel programming on shared memory systems, continues to extend its reach beyond pure HPC to include DSPs, real-time systems, and accelerators. The OpenMP API aims to provide high-level parallel language support for a wide range of applications, from automotive and aeronautics to biotech, automation, robotics and financial analysis.

New features in the OpenMP 4.0 API include:

• **Support for accelerators.** The OpenMP 4.0 API specification effort included significant participation by all the major vendors in order to support a wide variety of compute devices. OpenMP API provides mechanisms to describe regions of code where data and/or computation should be moved to another computing device. Several prototypes for the accelerator proposal have already been implemented.

• **SIMD constructs to vectorize both serial as well as parallelized loops.** With the advent of SIMD units in all major processor chips, portable support for accessing them is essential. OpenMP 4.0 API provides mechanisms to describe when multiple iterations of the loop can be executed concurrently using SIMD instructions and to describe how to create versions of functions that can be invoked across SIMD lanes.
July: Released OpenMP 4.0 after 5 years

“This represents collaborative work by many of the brightest in industry, research, and academia, building on the consensus of 26 members. *We strive to deliver high-level parallelism that is portable across 3 widely-implemented common General Purpose languages, productive for HPC and consumers, and delivers highly competitive performance.* I want to congratulate all the members for coming together to create such a momentous advancement in parallel programming, under such tight constraints and industry challenges. With this release, the OpenMP API will move immediately forward to the next release to bring even more usable parallelism to everyone. – Michael Wong
Compilers coming

- GCC close to complete, expected in mid 2014
- Intel 13.1 compiler supports Accelerators
- Clang support for OpenMP completed and released in Clang 3.4
- Cray, TI, IBM, Oracle coming online
Significant Paradigm Shift

“OpenMP feels like a new language.”
- The most pervasive parallelism
- Supporting HPC and consumers

OpenMP adoption scorecard.
- How are we doing?
- Learning, compilers, and books.

What’s next for OpenMP?
More usable parallelism

- With OpenMP 4.0, modern OpenMP code breaks from traditional Shared Memory Parallelism, and works on heterogeneous devices across different memory architecture

- Provide high level way of programming parallelism
  - And non-proprietary

- Still supports HPC, but now also consumer parallelism
  - Error Model, tasks dependency, accelerators
We’re All Learning OpenMP 4.0

- The world is still in the early stages of gaining field experience using the new language features, individually and together.
  - Includes everyone: OpenMP members, authors, developers.

- Example now published separately from spec
Using OpenMP: Implementations.

- Compilers: Several available now or soon (2014).
  - GNU/GCC mid 2014
  - Intel
  - Clang

- Tools: Rice University HPCTools Kit, IBM POMP support


- Tutorials: day long and in booth
- Books and articles coming
- *More videos posted based on SC and other talks*
- *Webinars in the plan*
OpenMP 4.0 Features

- Support for accelerators
- SIMD constructs to vectorize both serial as well as parallelized loops
- Error handling (cancel construct)
- Thread affinity
- Task groups and dependency
- Support for Fortran 2003
- User-defined reductions
- Sequentially consistent atomics
- `OMP_DISPLAY_ENV` environment variable
- Array section syntax for C/C++
OpenACC 1 compared to OpenMP 4.0 (by Dr. James Beyer)

**OpenACC 1**
- Parallel (offload)
  - Parallel (multiple “threads”)
- Kernels
- Data
- Loop
- Host data
- Cache
- Update
- Wait
- Declare

**OpenMP 4.0**
- Target
- Team/Parallel
- Target Data
- Distribute/Do/for/Simd
- Target Update
- Declare Target
Agenda

- The OpenMP ARB
- History of OpenMP
- OpenMP 4.0 Parallelism
- OpenMP Future Parallelism
- OpenMP new Mission Statement
OpenMP future features

- OpenMP Tools: Profilers and Debuggers
- Consumer style parallelism: event/async/futures
- Enhance Accelerator support
- Additional Looping constructs
- Transactional Memory, Speculative Execution
- Task Model refinements
- CPU Affinity
- Common Array Shaping
- Full Error Model
- Interoperability
- Rebase to new C/C++/Fortran Standards
Future OpenACC vs future OpenMP
(by Dr. James Beyer)

OpenACC2
- enter data
- exit data
- data api
- routine
- async wait
- parallel in parallel
- tile
- Linkable
- Device_type

OpenMP future
- Unstructured data environment
- declare target
- Parallel in parallel or team
- tile
- Linkable
- Device_type

Slide 22
Agenda

- The OpenMP ARB
- History of OpenMP
- OpenMP 4.0 Parallelism
- OpenMP Future Parallelism
- OpenMP new Mission Statement
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”
Marketing

- Larger booth with bigger tables, 10 stools
- Booth mini-talks, giveaways, cake, beer
- More talks and tutorials, speaker gifts
Brochures/Guides

- OMP Welcome Guide (for serious prospects)
- OMP Meetings and Email Lists (for new members)
- OMP Membership Brochure (given out at conferences)
- OMP FAQ (given out at conferences)
- OMP Schedule Flier (SC related talks and schedules)
- OpenMP brochure-SC13 (given out at SC 13)
OpenACC (Statement in 2011)

- A spinoff from 4 OpenMP Members
  - NVIDIA, PGI, Cray, CAPS
- To address immediate customer needs
- To hold the IP
- And to Beta test the OpenMP Accelerator implementation
- In time, will be folded back to OpenMP in some form
- We continue to aim to merge in future
External linkage

- Multicore association
  - For closely distributed systems, have API
  - They carry our brochures

- OpenFPGA
  - C-code->(impulse)->verilog
  - C-code->(impulse)->verilog->(VPR)->FPGA-image
  - C-code-with-OpenMP->(impulse-exploiting-openMP)->verilog->(VPR)->FPGA-image

- Standard C++ Foundation
  - I am serving as Director and Vice-President
Major Website update

- 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
Future meetings

- [http://openmp.org/calendar.html](http://openmp.org/calendar.html)
- Winter 2014: Jan 27-31, Intel Santa Clara
- Spring 2014: April, London, UK
- July/August: possible 3rd meeting
- Fall 2013: Sept/Oct, IWOMP 2014: Brazil

- 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 than 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
Feedback