2020 Collegeville Workshop on Scientific Software

Developer Productivity

July 21 – 23, 2020

Thanks to Our Sponsors

COLLEGE OF
Saint Benedict
Saint John's
UNIVERSITY

- Intel Corporation and The Mathworks, Inc.
  - Support to fund student staff
- College of Saint Benedict & Saint John's University
  - Home institution, host of Zoom resources
- Thank you!

intel®

MathWorks®

Meeting Purpose

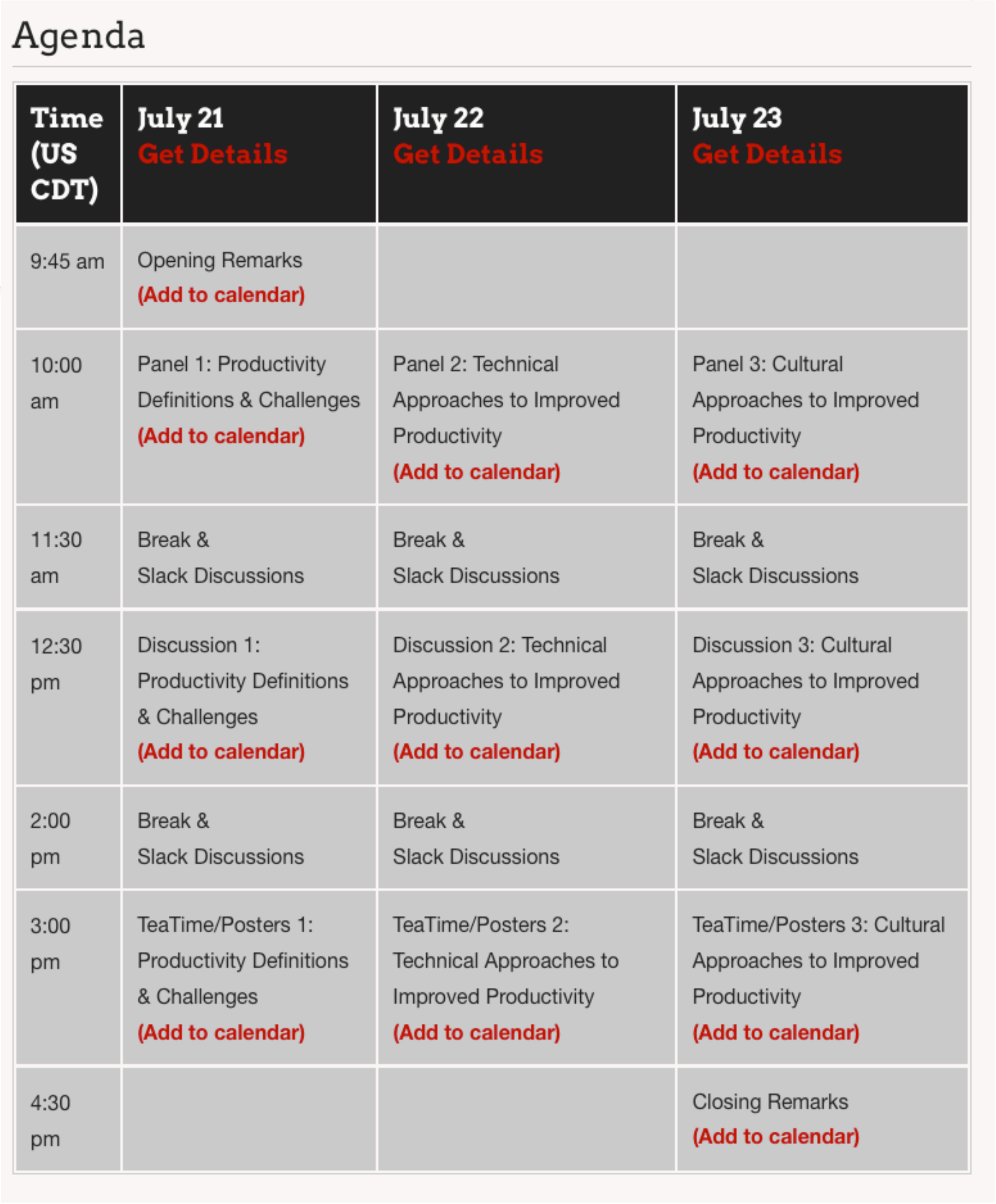
Code of Conduct

Be considerate, respectful, and collaborative. Communicate openly with respect for others, critiquing ideas rather than individuals. Avoid personal attacks directed toward other attendees, participants, and CW20 staff. Be mindful of your surroundings and of your fellow participants.

- Explore commonalities, differences, complementarities, workforces of
  - Academia
  - Industry
  - Labs
- Build knowledge and awareness to improve productivity
- Main purpose: Build community

Workshop Agenda

- All live sessions use Zoom
  - Panel: Single session
  - Discussion: Single session with random breakout assignments
  - TeaTime/Poster: Separate session for each
- Key Document: “Collegeville 2020 Main Page”
  - Google Doc
  - https://tinyurl.com/Collegeville2020
  - Single page to find out what is happening



Agenda

Time (US CDT)	July 21 Get Details	July 22 Get Details	July 23 Get Details
9:45 am	Opening Remarks (Add to calendar)		
10:00 am	Panel 1: Productivity Definitions & Challenges (Add to calendar)	Panel 2: Technical Approaches to Improved Productivity (Add to calendar)	Panel 3: Cultural Approaches to Improved Productivity (Add to calendar)
11:30 am	Break & Slack Discussions	Break & Slack Discussions	Break & Slack Discussions
12:30 pm	Discussion 1: Productivity Definitions & Challenges (Add to calendar)	Discussion 2: Technical Approaches to Improved Productivity (Add to calendar)	Discussion 3: Cultural Approaches to Improved Productivity (Add to calendar)
2:00 pm	Break & Slack Discussions	Break & Slack Discussions	Break & Slack Discussions
3:00 pm	TeaTime/Posters 1: Productivity Definitions & Challenges (Add to calendar)	TeaTime/Posters 2: Technical Approaches to Improved Productivity (Add to calendar)	TeaTime/Posters 3: Cultural Approaches to Improved Productivity (Add to calendar)
4:30 pm			Closing Remarks (Add to calendar)

Whitepapers & Recorded Content

- 29 videos on Collegeville YouTube Channel
  - 9 interviews with a variety of community members
  - 10 group discussion on productivity themes
  - 10 presentations from whitepapers

- 33 whitepapers on website
- Most have related video content

Make use of this content during and after the workshop

Use of Slack during meeting

- We encourage Slack chat during the meeting at any time
  - Especially during panels and breaks
  - To keep discussion organized around themes there is a special channel for each day (day-one-chat, day-two-chat, day-three-chat)
- Keep Zoom panel chat reserved for Q&A

Panel 2: Technical Approaches to Improved Productivity

- Panelists:
  - Vivek Sarkar, Georgia Tech
  - Jason Gates, Sandia National Labs
  - Charles Ferenbaugh, Los Alamos National Lab
  - Vadim Dyadechko, ExxonMobil
  - Anshu Dubey, Argonne National Lab
  - Hartwig Anzt, Karlsruhe Institute of Technology, U of Tennessee
- Moderator: Pat Quillen, MathWorks

Vivek Sarkar, Georgia Institute of Technology
Opening Remarks: Technical improvement approaches

- One or two high impact technical approaches to improve productivity
  1. Asynchronous tasks as a building block for heterogeneous and distributed parallelism
  2. Python based runtimes for productive medium-scale HPC and data analytics
- Key roles in defining and implementing the above approaches
  - Domain scientists, algorithm designers, application/framework/library developers
- Potential impact if approaches are successful
  - Significant improvement in productivity in use of current and future HPC hardware with extreme heterogeneity, especially for non-HPC experts
- Existing “bright spots” or other signs of progress
  - Asynchronous tasks are a building block of many parallel frameworks
  - New runtimes support distributed heterogeneous task parallelism with Python interfaces (e.g. Legate, Ray)
  - Demonstration of scalable multi-node multi-GPU performance with Python code containing NumPy and CuPy calls

Jason M. Gates Opening Remarks: Technical improvement approaches

- One or two high impact technical approaches to improve productivity
  - One script to rule them all: users, developers, & CI services all clone, configure, build, test, install the exact same way
  - Modular Jenkins Pipeline infrastructure to execute that script on supported machines
- Key roles in defining and implementing the above approaches
  - Project leads willing to devote the time to developing the DevOps infrastructure
  - Dedicated DevOps engineers to design, build, support, and extend that infrastructure
- Potential impact if approaches are successful
  - Improved replicability / decreased debugging time & frustration
  - Accelerated DevOps infrastructure development
- Existing “bright spots” or other signs of progress
  - Teams that have taken the plunge can speak to the payoff and convince others

Charles Ferenbaugh
Opening Remarks: Technical improvement approaches

- Approach to improve productivity: ***Robust testing system***
- Key roles:
  - Developer(s) to set up framework, make easy for other developers to use
  - Management support for the up-front time investment
- Potential impact
  - Early testing for correctness can greatly reduce errors/debugging later
- Existing “bright spots” or other signs of progress
  - Writing and running tests has become a regular part of developer workflow
    - Both for initial development and changes/refactoring
  - Some team members have found creative uses for the test system
    - Side-by-side runs for novel architecture vs. original versions
    - Test TPL integration independent of project code

Vadim: opening remarks: technical improvement approaches

- **High impact technical approaches to improve productivity:**
  - set right expectations: **stop looking for a silver bullet, make one little improvement a time**
  - **reduce non-essential complexity of your product/environment/processes**
  - **streamline routine tasks:** they should take zero effort and be human-proof
- **Key success factors in implementing the above approaches:**
  - raise awareness of your goals/motives/challenges to secure support
  - favor processes that motivate positive changes
- **Potential impact if approaches are successful:**
  - more transparent, robust, low-overhead operations
  - more time/opportunities for growing your product rather than fighting hiccups
- **Existing "bright spots" or other signs of progress:**
  - Git, Spack
  - growing awareness of complexity: **justified** refusal of feature requests / code sharing

Anshu Dubey Opening Remarks: Technical improvement approaches

- One or two high impact technical approaches to improve productivity
  - Testing harness that can be quickly customized by individual developers for specific code section
    - Downselect tests relevant to their development from the full test-suite
    - Enable running quickly on their development platform
    - Results viewable with the same user interface
- Key roles in defining and implementing the above approaches
  - Testing and verification lead of the project
- Potential impact if approaches are successful
  - Fewer interoperability bugs make their way into the main code base
- Existing “bright spots” or other signs of progress
  - We have used this method extensively in the past, and it is easy to implement if there is a will.

Hartwig Anzt Opening Remarks: Technical improvement approaches

- One or two high impact technical approaches to improve productivity
  - Clear software development cycle supported by scripts generating files and boilerplate code for extensions (create_algorithm convert_cuda_to_hip…)
  - Continuous Integration framework including exhaustive unit testing using googletest (95%)
  - Automated Docker image generation for continuous integration
- Key roles in defining and implementing the above approaches
  - Developer(s) to set up user-friendly framework and document the setup
  - Project lead willing to listen to the developers and provide resources needed
- Potential impact if approaches are successful
  - Reducing debugging, compilation issues, and hand-written boilerplate code
  - Lowering of the threshold for high performance implementation
  - Allowing developers to concentrate on hardware-specific performance optimization
- Existing “bright spots” or other signs of progress
  - Extending Ginkgo’s scope from NVIDIA to AMD GPUs took 4 weeks

Q&A Protocol

- Use Zoom chat to type your question (brief form)
- When your question is mentioned, unmute to ask verbally
- Unless asked to speak, please keep muted
- For general chat about panel, use Slack #day-two-chat
- Panel ends at 11:30 am CT or when no more questions
- Stopping screen sharing now so we can see faces ☺
- Will start up to give instructions for Discussions

Discussions, Posters, Teatimes Instructions

- Please sign in to next Zoom session promptly at 12:30 pm US CDT
- At 12:35 pm, we will use Zoom breakout rooms with automatic random assignments to form discussion groups of 6 – 8 people
- Discussions are scheduled from 12:30 – 2:00 pm CDT
- Posters and Teatime theme discussions start at 3:00 pm CDT
  - There are two posters and four teatime themes today
  - Each discussion has its own unique Zoom link (see Main Page for details)
  - Please visit any and all of them as you have time and interest
- We will not gather as one group until tomorrow morning's panel
- Questions?