2020 Collegeville Workshop on Scientific Software

Developer Productivity

July 21 – 23, 2020

Thanks to Our Sponsors

- 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!

Meeting Purpose

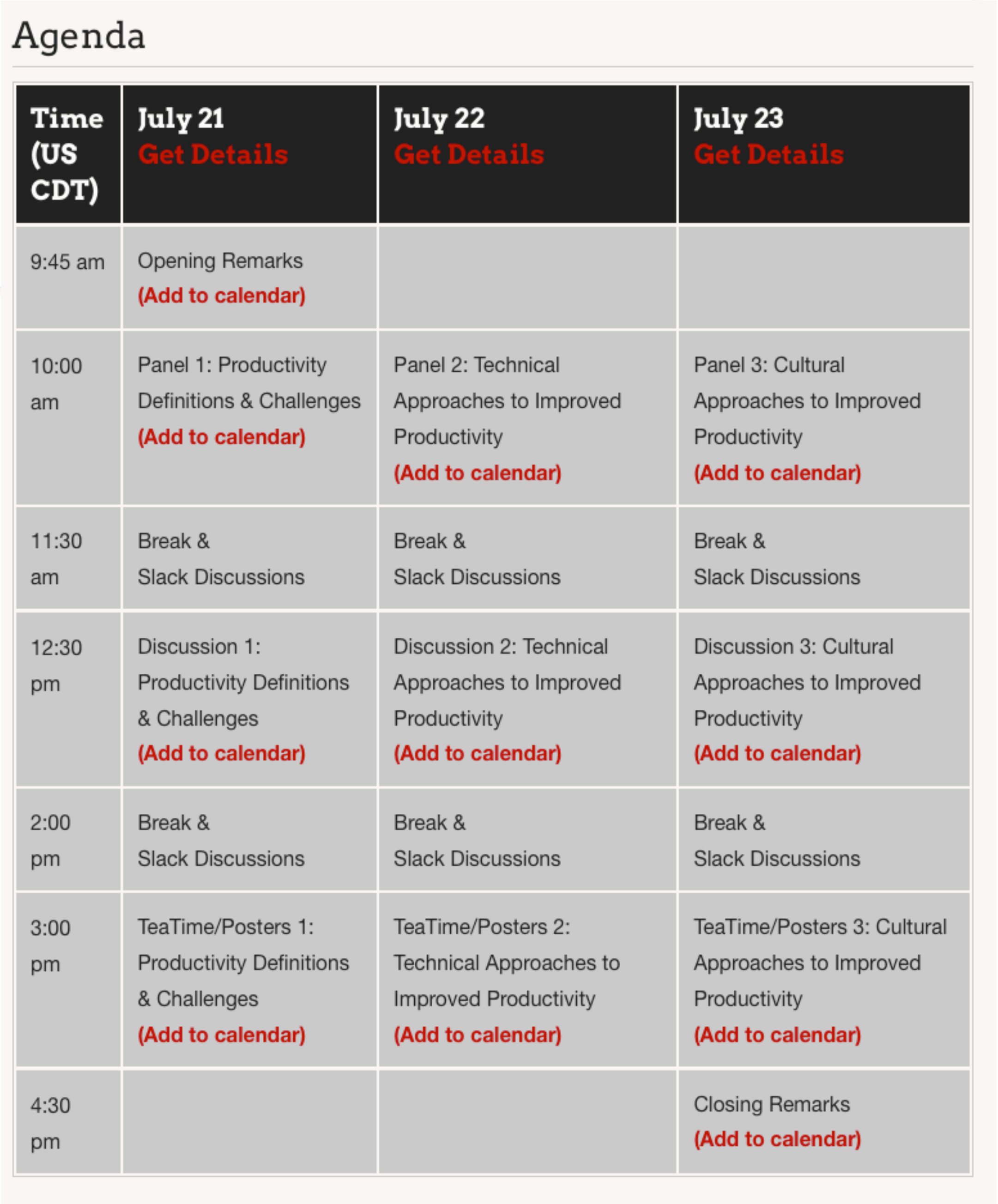
Code of Conduct

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 1: Productivity Definitions & Challenges

- Panelists:
  - Rick Arthur, GE
  - Sunita Chandrasekaran, U of Delaware
  - C. Fan Du, U of Texas
  - Hal Finkel, Argonne National Lab
  - Todd Gamblin, Lawrence Livermore National Lab
  - Theresa Windus, Iowa State
- Moderator: Mike Heroux, Sandia & St. John's

Rick Arthur Opening Remarks: Productivity definition and challenges

- *Definition attempt:* Minimizing the antipattern of avoidable waste while able to progress toward goals and exploit opportunities.
- LEAN/Agile “deadly sins” as source of *Productivity Challenges* → impact sponsor confidence & team morale
  - Overproduction/Inventory/WIP (leading to task switching, blocking dependencies, rework)
  - Defects/rework → quickly detect and fix before more dependency!
  - Task switching (due to oversight, mission multiplicity, bureaucracy) → aggregate and focus!
  - Dependencies/hand-offs/waiting (imbalanced scope/skills/resources) → planning
  - Adoption of “living” components w/ support ecosystem (e.g., E4S) in place of legacy or new redundant proprietary
  - High consequence / certification → monotonic process growth (nobody wants to sign off on relaxation)
    → rethink certification & VVUQ, engage regulators (collaborative vs. adversarial)
- Mitigation strategies: **Focus, Clarity & Empowerment** (“collaborative doers & deciders”)

Category	Doers answer these (clarify for Decider)	Decider can then give Doer
U (update)	What’s new? *(since last update)*	Confidence knowing progress made is reported
F (focus)	What’cha doin? *(now to next update)*	Empowerment through Clarity & Focus (*priority*) + alternatives
O (overflow)	How ‘bout when waiting or finishing early?	Backlog clarity, Focus/empowerment contingency / alternatives
D (distraction)	What’s been (or will) keep you from focusing?	Assistance / air cover: reduce *lurking surprise vs. expectations*
W (waiting)	What are you waiting on now or maybe soon?	Resources / authority or (*priority*) perspective

2020 Collegeville Workshop on Scientific Software: Panel on productivity definitions and challenges

Sunita Chandrasekaran Opening Remarks: Productivity definition and challenges

- Can someone use your software/code/package once you graduate and pick up from where you left?
- Key challenges you observe and experience and root causes
  - Algorithm is written to express the physics of the application but not for the hardware architecture
  - Unable to reproduce related work to evaluate against a novel work
  - Tried to solve a problem as one big piece instead of multiple small pieces
  - Did not test every step of the way, so unclear where and when the code broke
  - Not using version control – major issue
- Work you know or are doing to address these challenges
  - Regression testing, Jupyter Notebooks
  - Modular workflow
  - Automating several pieces of the workflow
  - DOCUMENTATION (caps is intentional :-))
  - Report bugs to vendors via ticket system
  - Provide vendors with workarounds to address a bug

C. Fan Du Opening Remarks: Productivity definition and challenges

- Interviews helped us identify three kinds of Ecosystem work:
  - Sensing: How is our software used, with what other components?
  - Adaptation: When nearby ecosystem changes, how do we adapt?
  - Synchronization: How can we be in sync with nearby ecosystem?
- Definitions:
  - Love feedback on these categories
  - How do projects account for these kinds of work? Any useful metrics?
- Key challenges:
  - How to motivate/incentive these kinds of work? Make visible?
  - e.g., Software publications are nice approaches for incentivizing sharing. But how do we incentivize work needed for maintainability and maintenance?

Hal Finkel's Opening Remarks: Productivity definition and challenges

- A working definition of productivity you use for making decisions
  - In the context of scientific programming: A metric inversely proportional to the total number of person hours necessary to plan, write, tune, debug, document, integrate, and maintain a piece of software.
- Key challenges you observe and experience and root causes
  - Insufficient tooling: We often lack good tools to efficiently program, debug, tune, and maintain our code. Tools for other domains don't apply to the programming languages, libraries, systems, or scale we require. We even lack good search tools to find relevant papers, code, and libraries.
  - Insufficient forethought: If only the "time to the first mostly-working version" is considered, bad things happen. If insufficient effort is given to prototyping and experimentation, bad things happen. Both the long-term and short-term aspects are important.
- Work you know or are doing to address these challenges
  - Work on compiler technology (e.g., our work in LLVM) to reduce abstraction penalties, scale debugging and tuning tools, automate refactoring (e.g., clang-tidy), semantic code search (e.g., some of the DARPA MUSE efforts), higher-level code generators (e.g., lbmpy).

Todd Gamblin Opening Remarks: Productivity definition and challenges

Programming productivity is a measure of how quickly one can go from a description to an implementation that satisfies the requirements.

- Language matters:
  - How simple and intuitive is the implementation of my problem?
  - What extrinsic (non-problem-related) concerns are handled for me?
  - Does the language meet the performance requirements?
- Terseness not always a win: complex language is hard to reason about, so simple systems can win.

- **Key challenges you observe and experience and root causes:**
  - Building HPC code is a very slow and unproductive task, due to:
    - Need for performance: de facto requirement to distribute as source code (lack of prebuilt binaries)
    - Complexity and number of supported platforms and environments
  - Optimizing HPC code is hard because it’s hard to share knowledge across environments
    - What’s good for one system/code/input deck may not be good for another
    - No standard way to track provenance across systems

- **Work you know or are doing to address these challenges**
  - DSLs for describing builds and configurations (Spack has: CLI DSL, package DSL, YAML for envs and configs)
  - Creating models of binary packages so that we can better understand compatibility and enable reuse
  - Addressing complexity with solvers & well-defined semantics
    - model and understand constraints → find good configurations

2020 Collegeville Workshop on Scientific Software: Panel on productivity definitions and challenges

Theresa Windus Opening Remarks: Productivity definition and challenges

- Fast development, accomplishes goals, quality, reusability, easy to build and debug, and sustainability (do we need them all?)
- What metrics to use to tell if you are doing a good job?
- Key challenges
    - No or little articulated design/documentation
    - Prototype becomes THE software
    - Lack of time/money/education
    - Lack of rewards for reuse, quality development
- Molecular Sciences Software Institute (MolSSI - https://molssi.org/) – through education, development of standards, awareness, and best practices
- CMakePP (https://github.com/CMakePP/)

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-one-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?