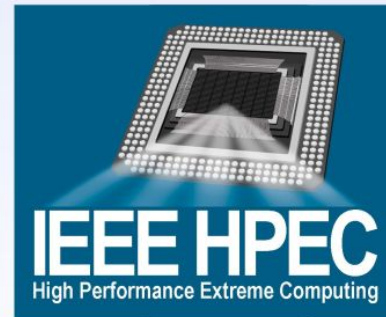


Welcome to Session 1-4:

Fastcode @

September 15, 2025



- Please make sure you are muted
- No recording or transcriptions/closed captioning
- Turn off your video (unless you are speaking) to save bandwidth
- Post questions and comments in Zoom chat

Task-Parallel Technology for Software Performance Engineering

Helping you select the most suitable task-parallel platform for your parallel workloads

Agenda

- 3:50PM Bruce Hoppe (10 minutes with Q&A at 4:45PM)
- 4:00PM TW Huang (15 minutes with Q&A at 4:45PM)
- 4:15PM Angelina Lee (15 minutes with Q&A at 4:45PM)
- 4:30PM Tim Mattson (15 minutes with Q&A at 4:45PM)
- 4:45PM **Discussion (including questions for any/all speakers)**
- 5:15PM Adjourn



Task-Parallel Technology for Software Performance Engineering

Helping you select the most suitable task-parallel platform for your parallel workloads

- Bruce Hoppe (MIT): *Fastcode: An Open-Source Community for Making Software Performance Engineering Easy and Fun*
- Tsung-Wei Huang (Univ. of Wisconsin): *Taskflow -- A General-Purpose Task-Parallel Programming System*
- I-Ting Angelina Lee (WUSTL): *OpenCilk -- A Modular and Extensible Software Infrastructure for Fast Task-Parallel Code*
- Tim Mattson (Human Learning Group): *Multithreaded Parallel Python Through OpenMP Support in Numba*



Fastcode

An open-source community
for making software performance engineering
easy and fun

Bruce Hoppe, PhD
Fastcode Community Manager



Software Performance Engineering (SPE)

Making software run fast or otherwise consume few resources such as time, storage, energy, network bandwidth, etc.

We have a choice: SPE can be tedious, expensive, haphazard, and controlled by “high priests”; or it can be fun, cheap, principled, and democratically available to the average programmer.

—Leiserson et al. (2023)



Fastcode

An open-source community for making SPE fun, cheap, principled, and democratically available to the average programmer

Four pillars

- *Research*: supporting the development of SPE as a principled scientific field
- *Education*: making it easier for instructors to teach SPE
- *Technology*: improving the tools that people use for SPE
- *Skill-building*: providing places for hands-on practicing of SPE skills.

<https://fastcode.org>



Fastcode

An open-source community for making SPE fun, cheap, principled, and democratically available to the average programmer

Four pillars

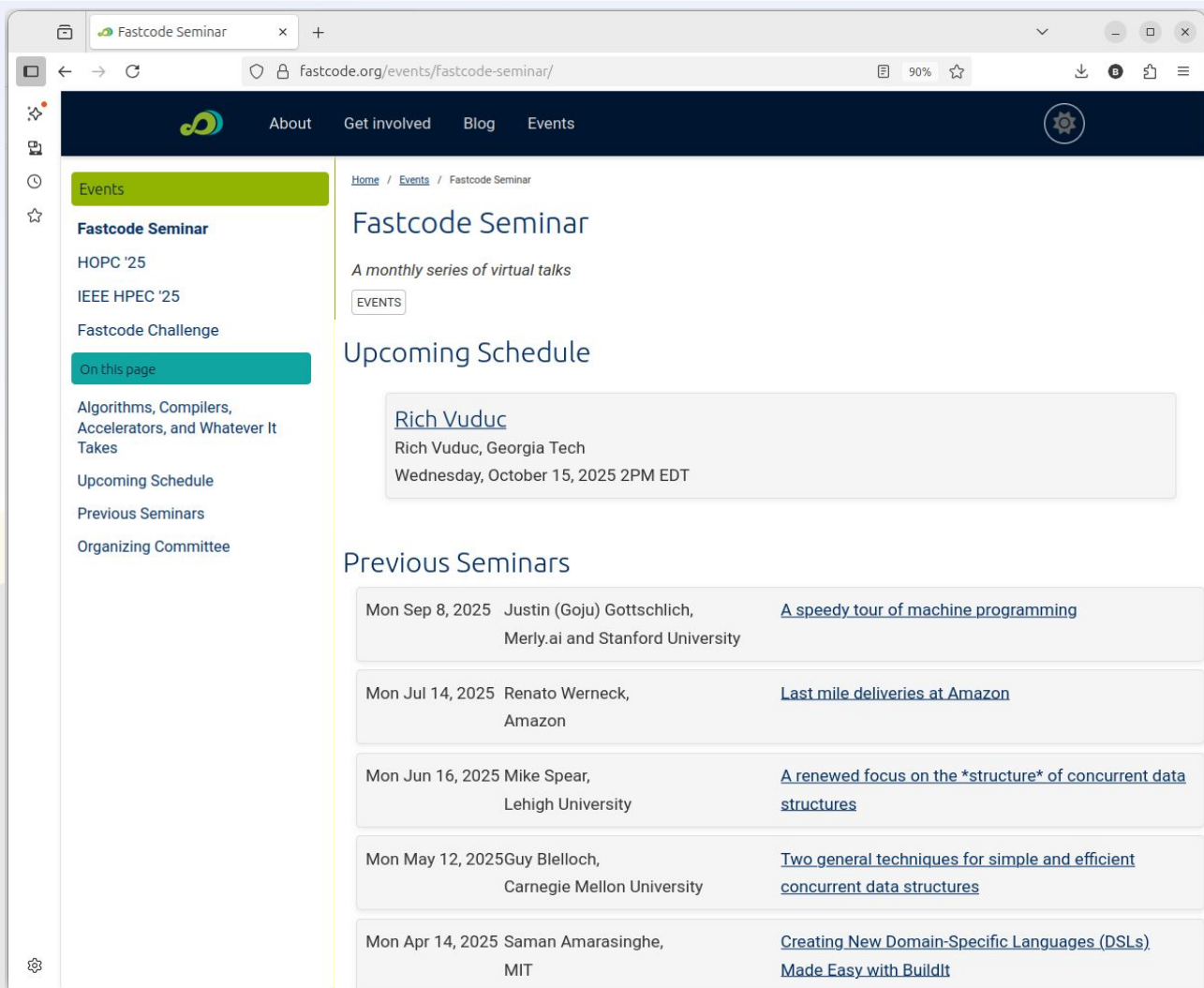
- *Research*: supporting the development of SPE as a principled scientific field
- *Education*: making it easier for instructors to teach SPE
- *Technology*: improving the tools that people use for SPE
- *Skill-building*: providing places for hands-on practicing of SPE skills.

<https://fastcode.org>



Free monthly seminar on “algorithms, compilers, accelerators, and whatever it takes”

<https://fastcode.org>

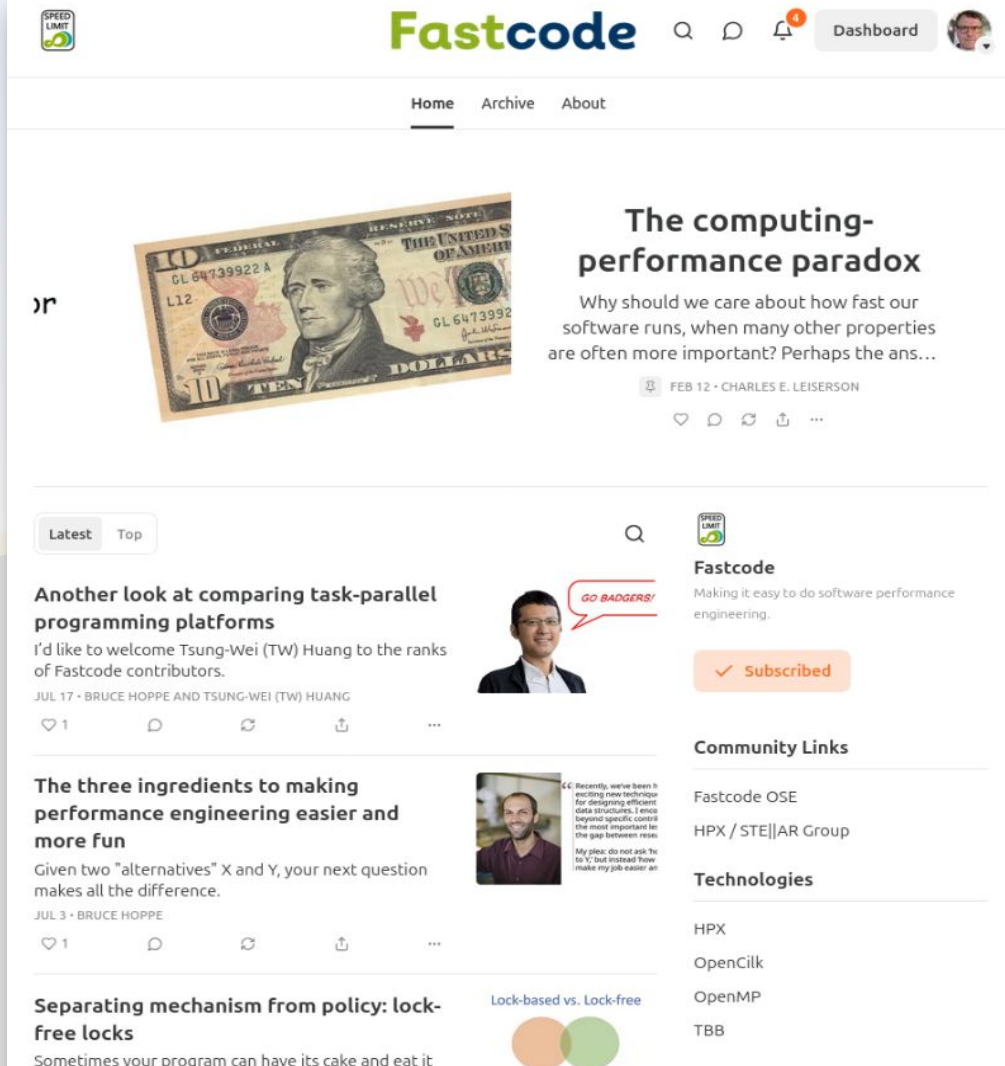


The screenshot shows the Fastcode Seminar website in a web browser. The browser's address bar displays `fastcode.org/events/fastcode-seminar/`. The website has a dark blue header with the Fastcode logo and navigation links: "About", "Get involved", "Blog", and "Events". A sidebar on the left contains a list of links: "Events" (highlighted in green), "Fastcode Seminar", "HOPC '25", "IEEE HPEC '25", "Fastcode Challenge", "On this page" (highlighted in teal), "Algorithms, Compilers, Accelerators, and Whatever It Takes", "Upcoming Schedule", "Previous Seminars", and "Organizing Committee". The main content area features the title "Fastcode Seminar" with the subtitle "A monthly series of virtual talks" and a button labeled "EVENTS". Below this is the "Upcoming Schedule" section, which lists a seminar by Rich Vuduc from Georgia Tech on Wednesday, October 15, 2025 at 2PM EDT. The "Previous Seminars" section lists five past events with their dates, speakers, affiliations, and links to their respective pages.





Date	Speaker	Affiliation	Topic
Mon Sep 8, 2025	Justin (Goju) Gottschlich,	Merly.ai and Stanford University	A speedy tour of machine programming
Mon Jul 14, 2025	Renato Werneck,	Amazon	Last mile deliveries at Amazon
Mon Jun 16, 2025	Mike Spear,	Lehigh University	A renewed focus on the *structure* of concurrent data structures
Mon May 12, 2025	Guy Blelloch,	Carnegie Mellon University	Two general techniques for simple and efficient concurrent data structures
Mon Apr 14, 2025	Saman Amarasinghe,	MIT	Creating New Domain-Specific Languages (DSLs) Made Easy with BuildIt

Fastcode Blog

<https://fastcode.substack.com>



The screenshot shows the Fastcode blog homepage. At the top is a navigation bar with the Fastcode logo, search, and social media icons, and a 'Dashboard' button with a user profile picture. Below the navigation bar are links for 'Home', 'Archive', and 'About'. The main content area features a large article titled 'The computing-performance paradox' by Charles E. Leiserson, dated February 12. The article includes a thumbnail image of a US ten-dollar bill. Below this are two more article previews: 'Another look at comparing task-parallel programming platforms' by Bruce Hoppe and Tsung-Wei (TW) Huang, and 'The three ingredients to making performance engineering easier and more fun' by Bruce Hoppe. On the right side, there is a sidebar with the Fastcode logo, a 'Subscribed' button, and a 'Community Links' section listing 'Fastcode OSE', 'HPX / STE||AR Group', and 'Technologies' (HPX, OpenCilk, OpenMP, TBB). A Venn diagram titled 'Lock-based vs. Lock-free' is partially visible at the bottom right.






Fastcode    [Dashboard](#) 

[Home](#) [Archive](#) [About](#)

The computing-performance paradox

Why should we care about how fast our software runs, when many other properties are often more important? Perhaps the ans...

FEB 12 • CHARLES E. LEISERSON






    

[Latest](#) [Top](#)

Another look at comparing task-parallel programming platforms

I'd like to welcome Tsung-Wei (TW) Huang to the ranks of Fastcode contributors.






JUL 17 • BRUCE HOPPE AND TSUNG-WEI (TW) HUANG

 1    

The three ingredients to making performance engineering easier and more fun

Given two "alternatives" X and Y, your next question makes all the difference.


JUL 3 • BRUCE HOPPE




 1    

Separating mechanism from policy: lock-free locks


Sometimes your program can have its cake and eat it

[Lock-based vs. Lock-free](#)



Fastcode   

Making it easy to do software performance engineering.

 **Subscribed**

Community Links

[Fastcode OSE](#)

[HPX / STE||AR Group](#)

Technologies

[HPX](#)

[OpenCilk](#)

[OpenMP](#)

[TBB](#)

Fastcode Blog

Inspiration for today's session

<https://fastcode.substack.com>

Fastcode [Dashboard](#)

[Home](#) [Archive](#) [About](#)

The computing-performance paradox
Why should we care about how fast our software runs, when many other properties are often more important? Perhaps the ans...
FEB 12 • CHARLES E. LEISERSON

[Latest](#) [Top](#)

Another look at comparing task-parallel programming platforms
I'd like to welcome Tsung-Wei (TW) Huang to the ranks of Fastcode contributors.
JUL 17 • BRUCE HOPPE AND TSUNG-WEI (TW) HUANG
 1

The three ingredients to making performance engineering easier and more fun
Given two "alternatives" X and Y, your next question makes all the difference.
JUL 3 • BRUCE HOPPE
 1

Separating mechanism from policy: lock-free locks
Sometimes your program can have its cake and eat it
Lock-based vs. Lock-free

Fastcode
Making it easy to do software performance engineering.
[✓ Subscribed](#)

Community Links
[Fastcode OSE](#)
[HPX / STE||AR Group](#)

Technologies
[HPX](#)
[OpenCilk](#)
[OpenMP](#)
[TBB](#)



Finding performance-engineering El Dorado

When we compare programming platforms, we are trying to choose one that makes it as easy as possible to write good and efficient code. Charles Leiserson posted here that [performance is like currency](#), so we picture our comparison efforts as hacking through the jungle on a quest for [El Dorado](#), the lost city of gold, where money/performance comes freely to anyone who wants it.

Fastcode's feature comparison table of TPPs for SPE

	API Features											Runtime			
Platform	Communication Model	Distributed Memory	Heterogeneity	Graph Structure	Task Partitioning	Result Handling	Worker Management	Resilience Management	Work Mapping	Synchronization	Technological Readiness	Implementation Type	Data Distribution	Scheduling Methods (sm)	Performance Monitoring
C++ STL	sm			dag		a/m	auto		a/m	m	9	lib.			

Task-parallel technology | x

fastcode.org/about/task-parallel-tech/



90%

☆

ⓘ

🔍

☰


[About](#)
[Get involved](#)
[Blog](#)
[Events](#)


About

About SPE

Get started with SPE

Task-parallel tech for SPE

SPE Glossary

About Fastcode

On this page

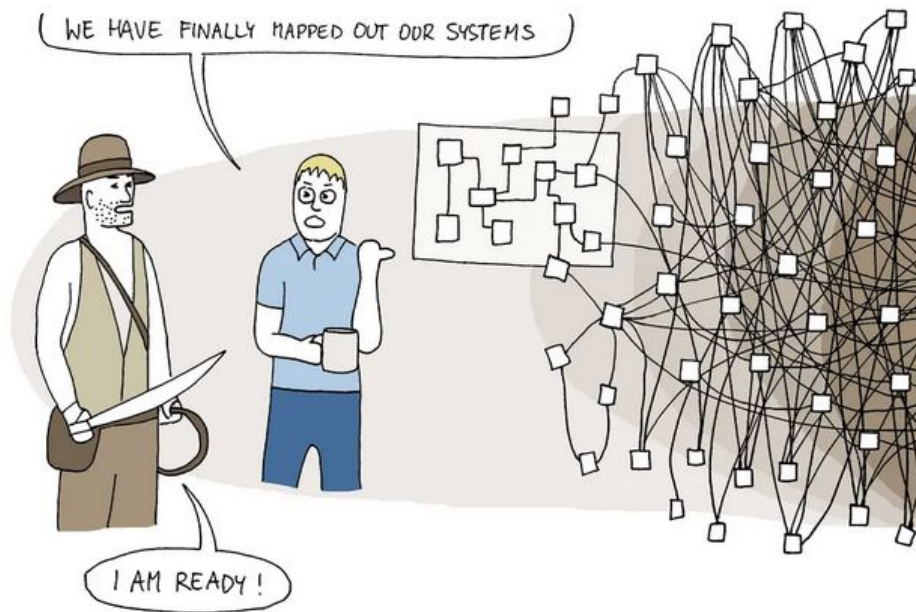
Feature comparison table

Feature comparison table

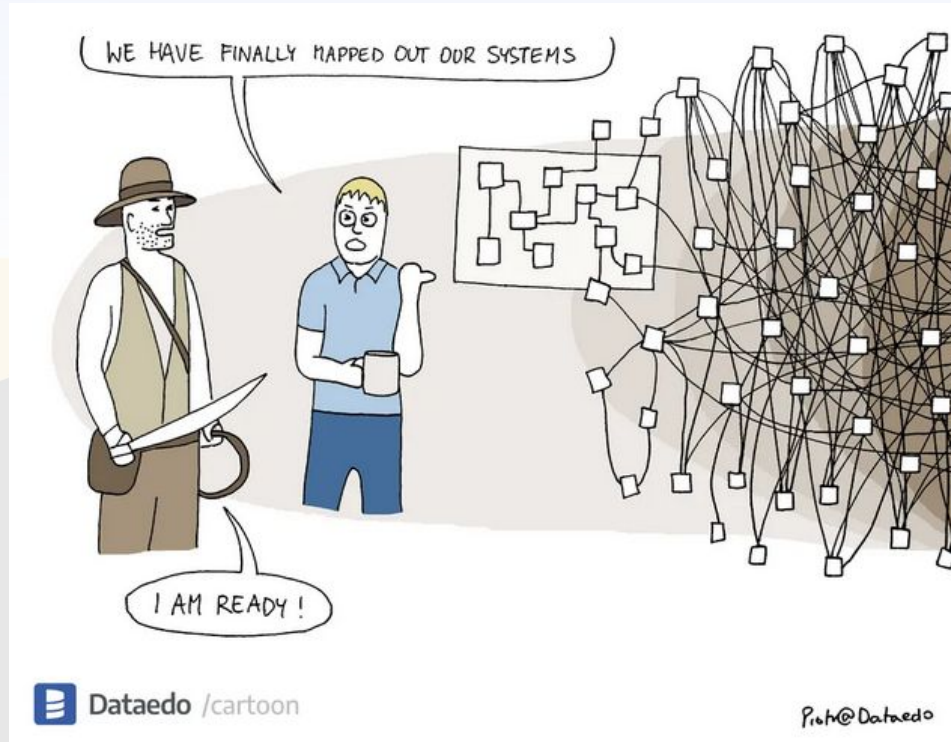
The following table is based on [A taxonomy of task-based parallel programming technologies for high-performance computing](#) by Thoman et al. With the authors' permission, we present an updated version of their classification, slightly abridged to focus on shared memory architecture. Our table adds two platforms: OpenCilk and Taskflow. (OpenCilk replaces Cilk Plus in their classification, and Taskflow is new.)

Platform	API Features											Runtime			
	Communication Model	Distributed Memory	Heterogeneity	Graph Structure	Task Partitioning	Result Handling	Worker Management	Resilience Management	Work Mapping	Synchronization	Technological Readiness	Implementation Type	Data Distribution	Scheduling Methods (sm)	Performance Monitoring
C++ STL	sm			dag		a/m	auto		a/m	m	9	lib.			
TBB	sm			tree		auto	auto		auto	auto	8	lib.		ws	off
OpenMP	sm		auto	dag		auto	m		auto	a/m	9	ext.		mult	off/on
OmpSs	sm		auto	dag		auto	auto	✓	auto	a/m	5	ext.	auto	mult	off/on
OpenCilk	sm			dag	✓	auto	auto		auto	a/m	8	lang.		ws	off
Taskflow	sm	auto	m	dag		auto	auto		auto	a/m	8	lib.		ws	off
HPX	gas	auto	m	dag	✓	m	a/m		a/m	m	6	lib.	auto	mult	off/on

"Searching for El Dorado" meets actual jungle



What's happening in your jungle?



Task-Parallel Technology for Software Performance Engineering

Helping you select the most suitable task-parallel platform for your parallel workloads

- Bruce Hoppe (MIT): *Fastcode: An Open-Source Community for Making Software Performance Engineering Easy and Fun*
- Tsung-Wei Huang (Univ. of Wisconsin): *Taskflow -- A General-Purpose Task-Parallel Programming System*
- I-Ting Angelina Lee (WUSTL): *OpenCilk -- A Modular and Extensible Software Infrastructure for Fast Task-Parallel Code*
- Tim Mattson (Human Learning Group): *Multithreaded Parallel Python Through OpenMP Support in Numba*



Task-Parallel Technology for Software Performance Engineering

Helping you select the most suitable task-parallel platform for your parallel workloads

- Bruce Hoppe (MIT): *Fastcode: An Open-Source Community for Making Software Performance Engineering Easy and Fun*
- Tsung-Wei Huang (Univ. of Wisconsin): *Taskflow -- A General-Purpose Task-Parallel Programming System*
- I-Ting Angelina Lee (WUSTL): *OpenCilk -- A Modular and Extensible Software Infrastructure for Fast Task-Parallel Code*
- Tim Mattson (Human Learning Group): *Multithreaded Parallel Python Through OpenMP Support in Numba*



Task-Parallel Technology for Software Performance Engineering

Helping you select the most suitable task-parallel platform for your parallel workloads

- Bruce Hoppe (MIT): *Fastcode: An Open-Source Community for Making Software Performance Engineering Easy and Fun*
- Tsung-Wei Huang (Univ. of Wisconsin): *Taskflow -- A General-Purpose Task-Parallel Programming System*
- I-Ting Angelina Lee (WUSTL): *OpenCilk -- A Modular and Extensible Software Infrastructure for Fast Task-Parallel Code*
- Tim Mattson (Human Learning Group): *Multithreaded Parallel Python Through OpenMP Support in Numba*



Discussion

*Selecting the most suitable
task-parallel platform for your
parallel workloads*



Next at IEEE-HPEC 2025

- ▶ Session 1-5 starts at 5:30pm ET
- ▶ Go back to [Engagez.net/HPEC2025](https://engagez.net/HPEC2025) in your browser
- ▶ Look up next session in Sessions page
- ▶ Go to Breaks and Meetups to meet others at conference
- ▶ Breakout Rooms available for further discussion
- ▶ Numbered 1-4 – in order of talks
- ▶ Session Host will pass on Zoom Host role to those still in discussion

Questions about IEEE-HPEC?

Please go to engagez.net/HPEC2025

Click on Forums and Meetups and submit question to the Help-Desk forum