Welcome to Session 1-4:

Fastcode @ IEEE HPEC

September 15, 2025

High Performance Extreme Computing

- 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

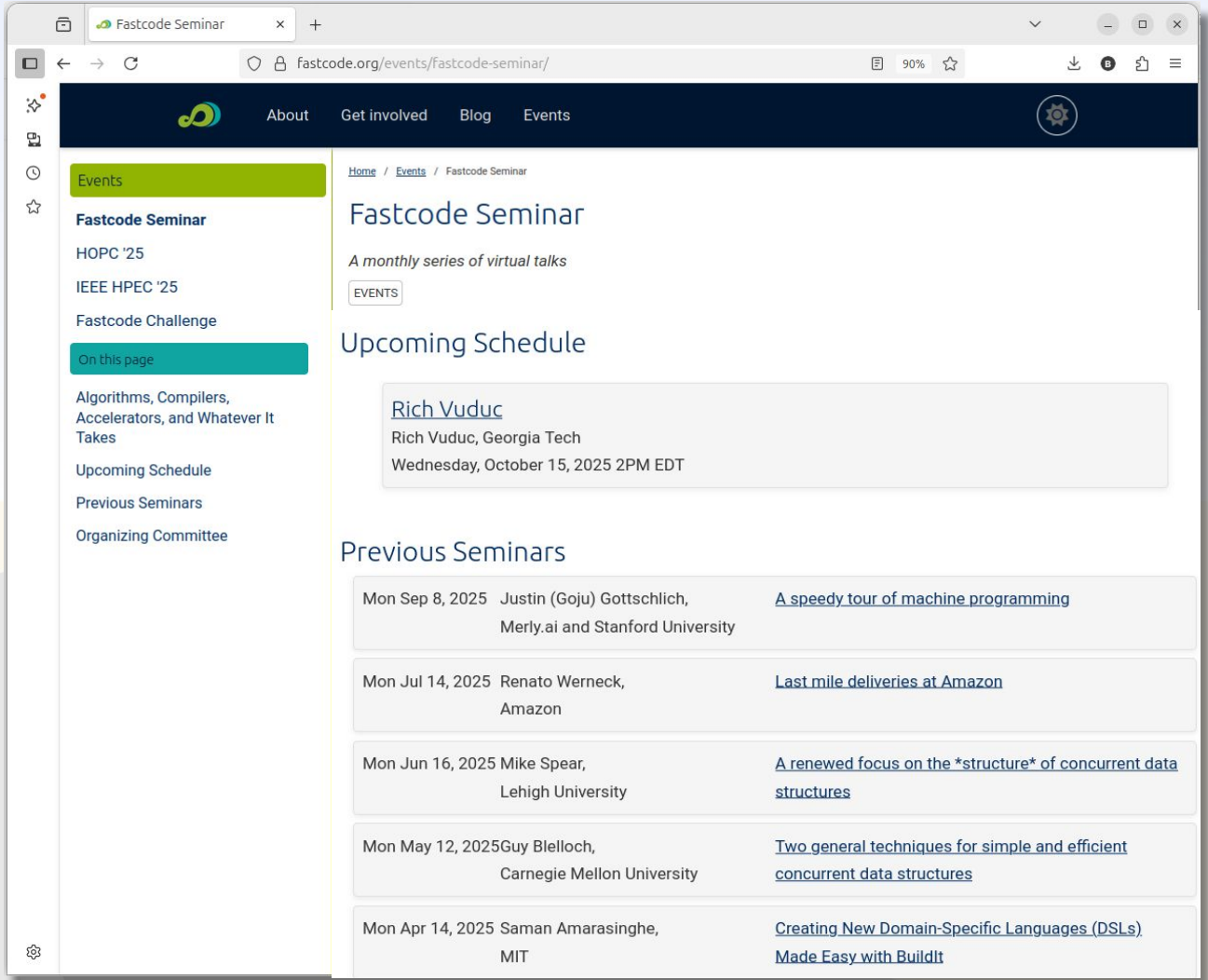
Four pillars

- *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

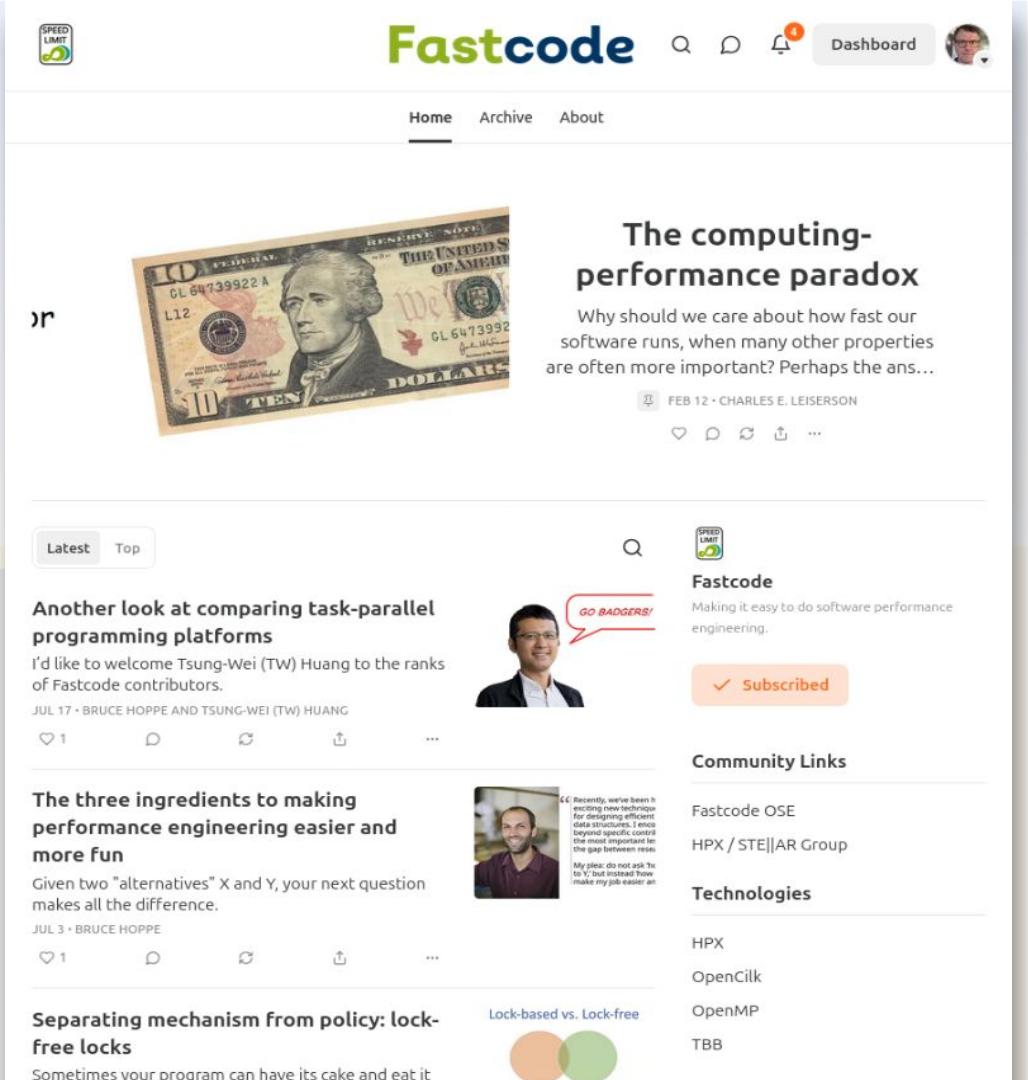
Free monthly seminar on “algorithms, compilers, accelerators, and whatever it takes”

https://fastcode.org



Fastcode Blog

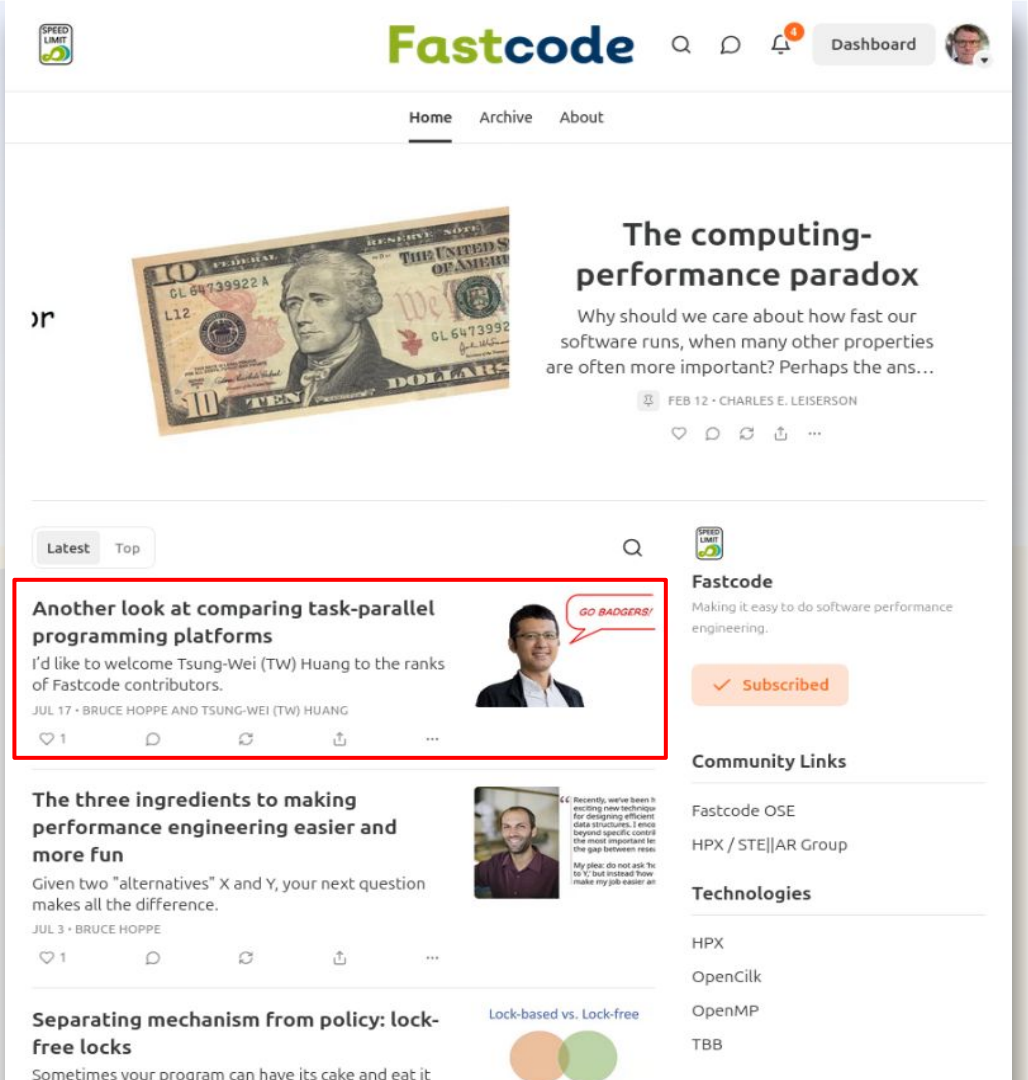
https://fastcode.substack.com



Fastcode Blog

Inspiration for today's session

https://fastcode.substack.com



Another look at comparin

fastcode.substack.com/p/updating-our-comparison-of-task-parallel

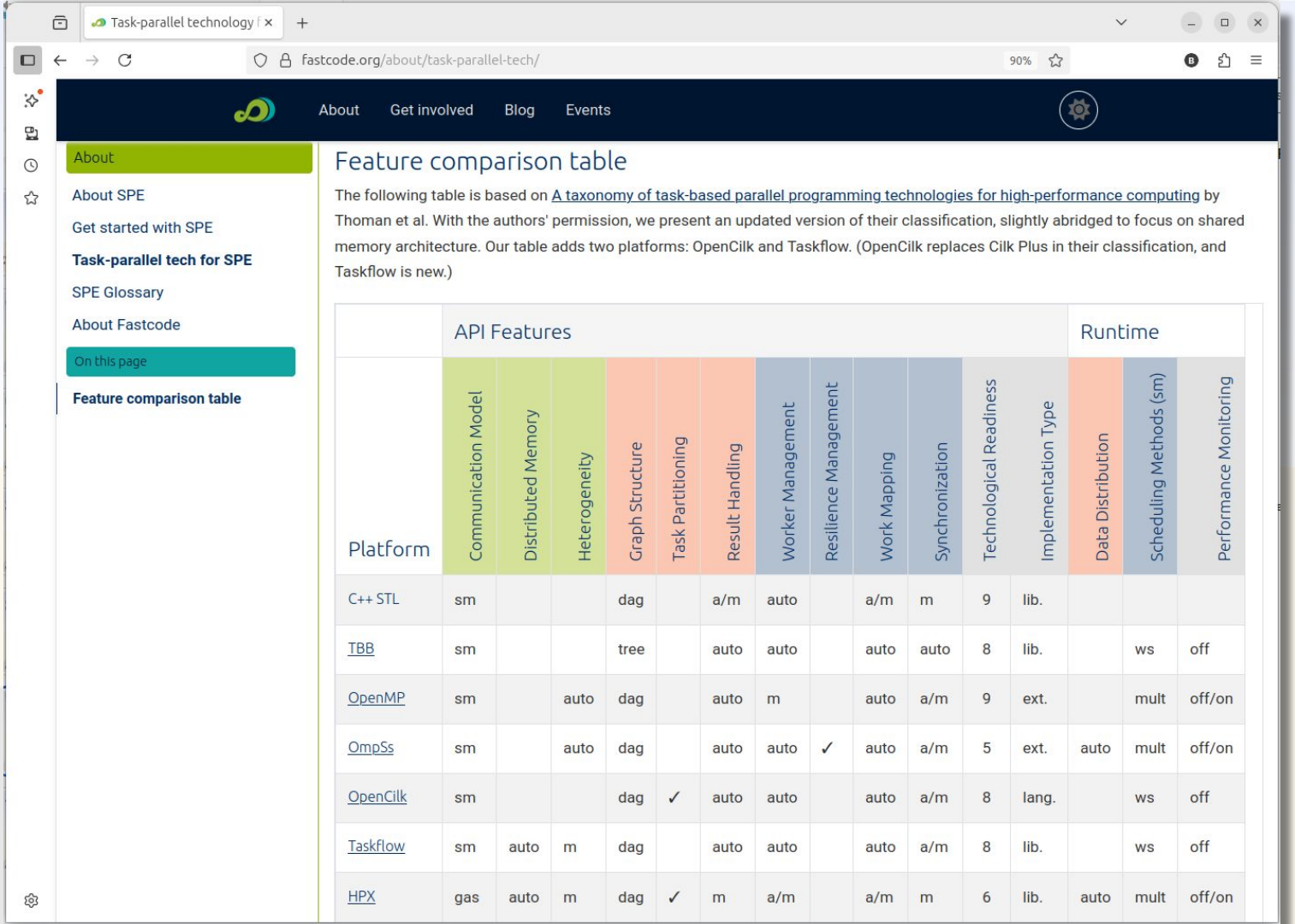
Fastcode

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.			



Feature comparison table

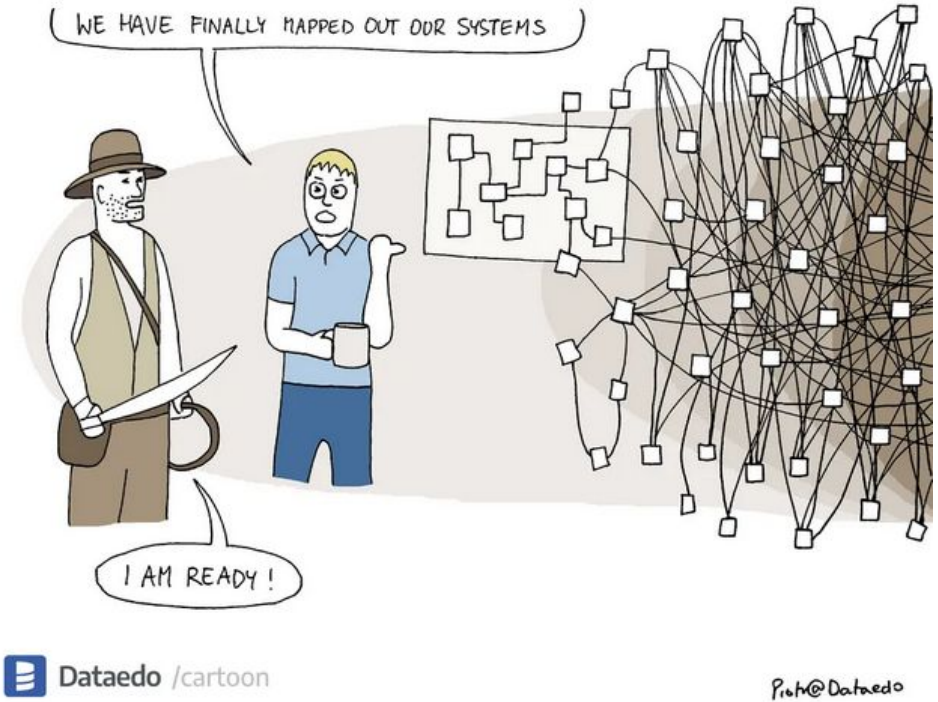
The following table is based on A taxonomy of task-based parallel programming technologies for high-performance computing by Thoman et al. With the authors' permission, we present an updated version of their classification, slightly abridged to focus on shared memory architecture. Our table adds two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

	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.			
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

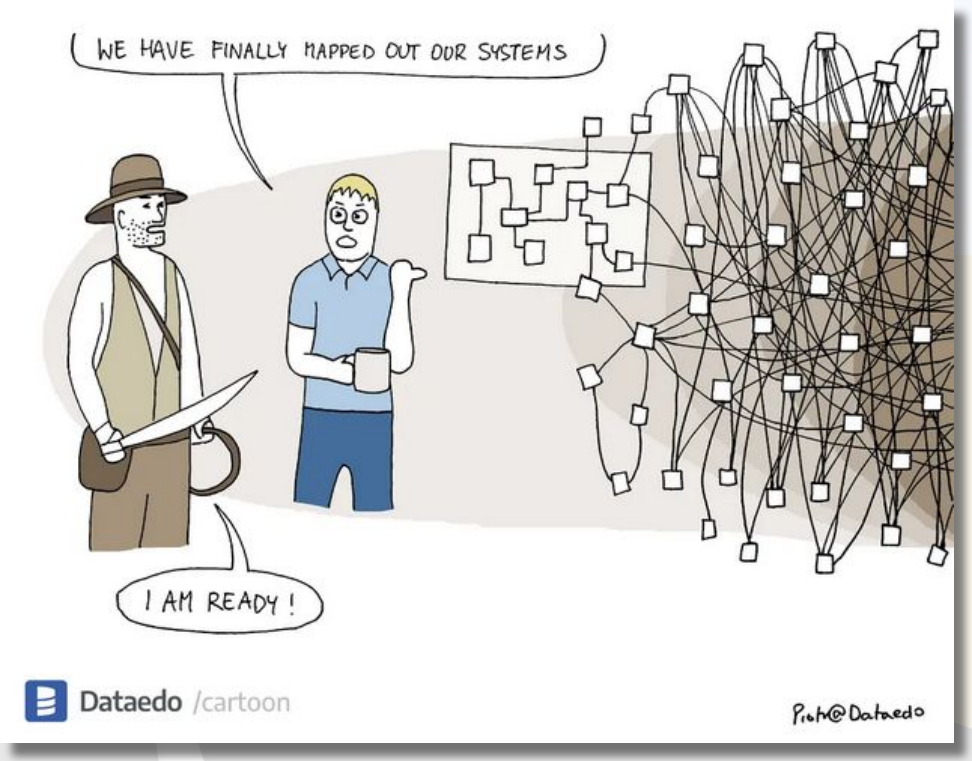
Another look at comparin

fastcode.substack.com/p/updating-our-comparison-of-task-parallel

“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 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