

# OpenMP<sup>®</sup>

## SC25 OpenMP Tech Talk Series



Why we need OpenMP  
more than ever

Tim Mattson | Human Learning Group

# Why we need OpenMP more than ever

Tim Mattson, Human Learning Group



# The modern era of parallelism

**Parallelism has been around since the early 60's  
(or perhaps earlier).**

**... the nature of parallelism and its impact on HPC  
changed when the focus shifted to parallel systems  
built from Commercial Off The Shelf (COTS)  
microprocessors**



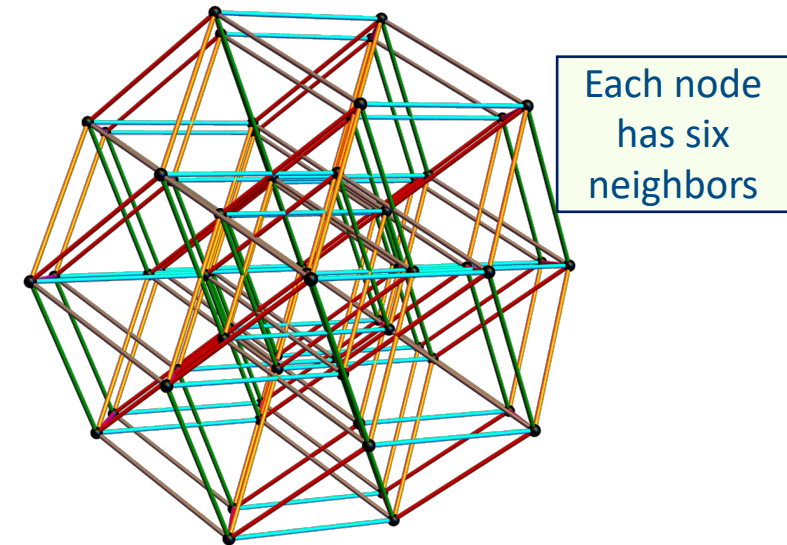


# The modern era of parallelism: It all started at Caltech



- The **Caltech Cosmic Cube** developed by Charles Seitz and Geoffrey Fox in 1981

- 64 Intel 8086/8087 processors
- 128kB of memory per processor
- 6-dimensional hypercube network<sup>^</sup>



- COTS processors radically cut HPC hardware costs.
- COTS enabled clusters so now any university could have an HPC system.

The cosmic cube, Charles Seitz, Communications of the ACM, Vol 28, number 1 January 1985, p. 22



Eugene Brooks\*  
famously summarized  
the impact of COTS  
microprocessors on  
supercomputing with  
the phrase ... The  
attack of the killer  
micros

\*Eugene Brooks, SC'90



Movie Poster by google gemini



# With so many parallel systems, people went wild and created numerous parallel programming environments ...

## Parallel programming environments in the 90's

ABCPL	Cashmere	DICE	GLU	ISETL-Linda	Nano-Threads	Parafrese2	P-RIO	Split-C.
ACE	C4	DIPC	GUARD	ParLin	NESL	Paralation	Prospero	SR
ACT++	CC++	DOLIB	HASL	Eilean	NetClasses++	Parallel-C++	Proteus	Sthreads
Active messages	Chu	DOME	Haskell	P4-Linda	Nexus	Parallaxis	QPC++	Strand.
Ada	Charlotte	DOSMOS.	HPC++	Glenda	Nimrod	ParC	PVM	SUIF.
Adl	Charm	DRL	HPF	POSYBL	NOW	ParLib++	PSI	Synergy
Adsmith	Charm++	DSM-Threads	JAVAR.	Objective-Linda	Nx	ParLin	PSDM	Telegrphos
ADDAP	Cid	Ease .	HORUS	LiPS	Objective	Parmacs	Quake	SuperPascal
AFAPI	Cilk	ECO	HPC	Locust	Linda	Parti	Quark	TCGMSG.
ALWAN	CM-Fortran	Eiffel	IMPACT	Lparx	Occam	pC	Quick	Threads.h++.
AM	Converse	Eilean	ISIS.	Lucid	Omega	pC++	Threads	TreadMarks
AMDC	Code	Emerald	JAVAR	Maisie	OpenMP	PCN	Sage++	TRAPPER
AppLeS	COOL	EPL	JADE	Manifold Mentat	Orca	PCP	SCANDAL	uC++
Amoeba	CORRELATE	Excalibur	Java RMI	Legion	OOF90	PH	SAM pC++	UNITY
ARTS	CparPar	Express	javaPG	Meta Chaos	P++	PEACE	SCHEDULE	UC
Athapascan-0b	CPS	Falcon	JavaSpace	Midway	P3L	PCU	SciTL	V
Aurora	CRL	Filaments	JIDL	Millipede	P4	PET	POET	ViC*
Automap	CSP	FM	Joyce	Mirage	P4-Linda	PETSc	SDDA	Visifold V-NUS
bb_threads	Cthreads	FLASH	Khoros	MpC	Pablo	PENNY	SHMEM	VPE
Blaze	CUMULVS	The FORCE	Karma	MOSIX	PADE	Phosphorus	SIMPLE	Win32 threads
BSP	DAGGER	Fork	KOAN/Fortran-S	Modula-P	PADRE	POET.	Sina	WinPar
BlockComm	DAPPLE	Fortran-M	LAM	Modula-2*	Panda	Polaris	SISAL.	WWWinda
C*	Data Parallel C	FX	Lilac	Multipol	Papers	POOMA	distributed	XENOOPS
C* in C	DC++	GA	Linda	MPI	AFAPI.	POOL-T	smalltalk	XPC
C**	DCE++	GAMMA	JADA	MPC++	Para++	PRESTO	SML.	Zounds
CarLOS	DDD	Glenda	WWWinda	Munin	Paradigm		SONiC	ZPL

This list was compiled by looking at papers listed in conference proceedings from the mid to the late 90's.





# With so many parallel systems, people went wild and created numerous parallel programming environments ...

## Parallel programming environments in the 90's

ABCPL	Cashmere	DICE	GLU	ISETL-Linda	Nano-Threads	Parafrase2	P-RIO	Split-C.
ACE	C4	DIPC	GUARD	ParLin	NESL	Paralation	Prospero	SR
ACT++	CC++	DOLIB	HAsL.	Eilean	NetClasses++	Parallel-C++	Proteus	Sthreads
Active messages	Chu	DOME	Haskell	P4-Linda	Nexus	Parallaxis	QPC++	Strand.
Ada	Charlotte	DOSMOS.	HPC++	Glenda	Nimrod	ParC	PVM	SUIF.
Adl	Charm	DRL	HPF	POSYBL	NOW	ParLib++	PSI	Synergy
Adsmith	Charm++	DSM-Threads	JAVAR.	Objective-Linda	Nx	ParLin	PSDM	Telegrphos
ADDAP	Cid	Ease .	HORUS	LiPS	Objective	Parmacs	Quake	SuperPascal

### 2 Problems emerged from this overabundance abundance of parallel programming environments

- Application developers had to waste time figuring out which programming environments to use ... then once the choice was made, wasted time porting to new prog-envs when new hardware demanded it.
- Vendors obey key customers ... and had to support multiple prog envs ... engineering is a zero-sum game. Time spent on new prog envs means less time to make the core prog envs more stable.

Automap	CSP	FM	Joyce	Mirage	P4-Linda	PETSc	SDDA	Visifold V-NUS
bb_threads	Cthreads	FLASH	Khoros	MpC	Pablo	PENNY	SHMEM	VPE
Blaze	CUMULVS	The FORCE	Karma	MOSIX	PADE	Phosphorus	SIMPLE	Win32 threads
BSP	DAGGER	Fork	KOAN/Fortran-S	Modula-P	PADRE	POET.	Sina	WinPar
BlockComm	DAPPLE	Fortran-M	LAM	Modula-2*	Panda	Polaris	SISAL.	WWWinda
C*	Data Parallel C	FX	Lilac	Multipol	Papers	POOMA	distributed	XENOOOPS
C* in C	DC++	GA	Linda	MPI	AFAPI.	POOL-T	smalltalk	XPC
C**	DCE++	GAMMA	JADA	MPC++	Para++	PRESTO	SML.	Zounds
CarLOS	DDD	Glenda	WWWinda	Munin	Paradigm		SONiC	ZPL

This list was compiled by looking at papers listed in conference proceedings form the mid to the late 90's.



**Then a brief moment of sanity  
prevailed**





# Birth of MPI

Workstation vendors wanted into the HPC market

Needed a production worthy, full-featured system portable across different clusters

## Hardware:

By the early 90's, massively parallel processors (MPPs) and the new trend with clusters convinced even the skeptics that the "killer micros" had won.

MPP Vendors

Needed a common foundation to build a parallel SW industry

User Community

Fed-up recoding as they moved between platforms

After several years of informal discussions, the MPI forum was created in 1992. A draft specification was presented one year later at SC'93.

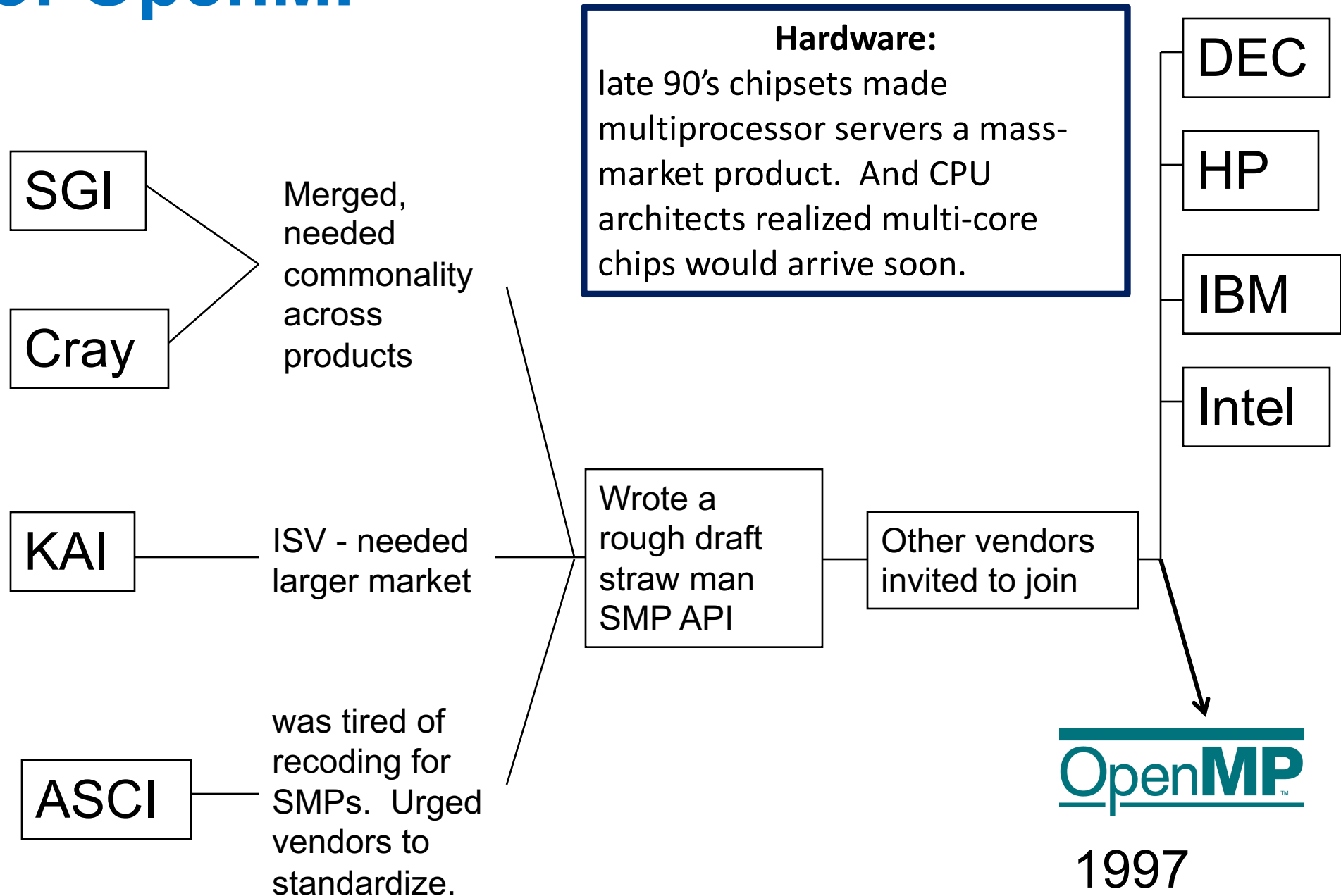


1994

Many of us worked in the MPI forum ... leadership came from the DOE National Labs. In particular, the reference implementation from Bill Gropp and Rusty Lusk of Argonne national lab called MPIch helped us get it right in the 1.0 specification and made sure a working implementation of the standard was available right from the beginning.



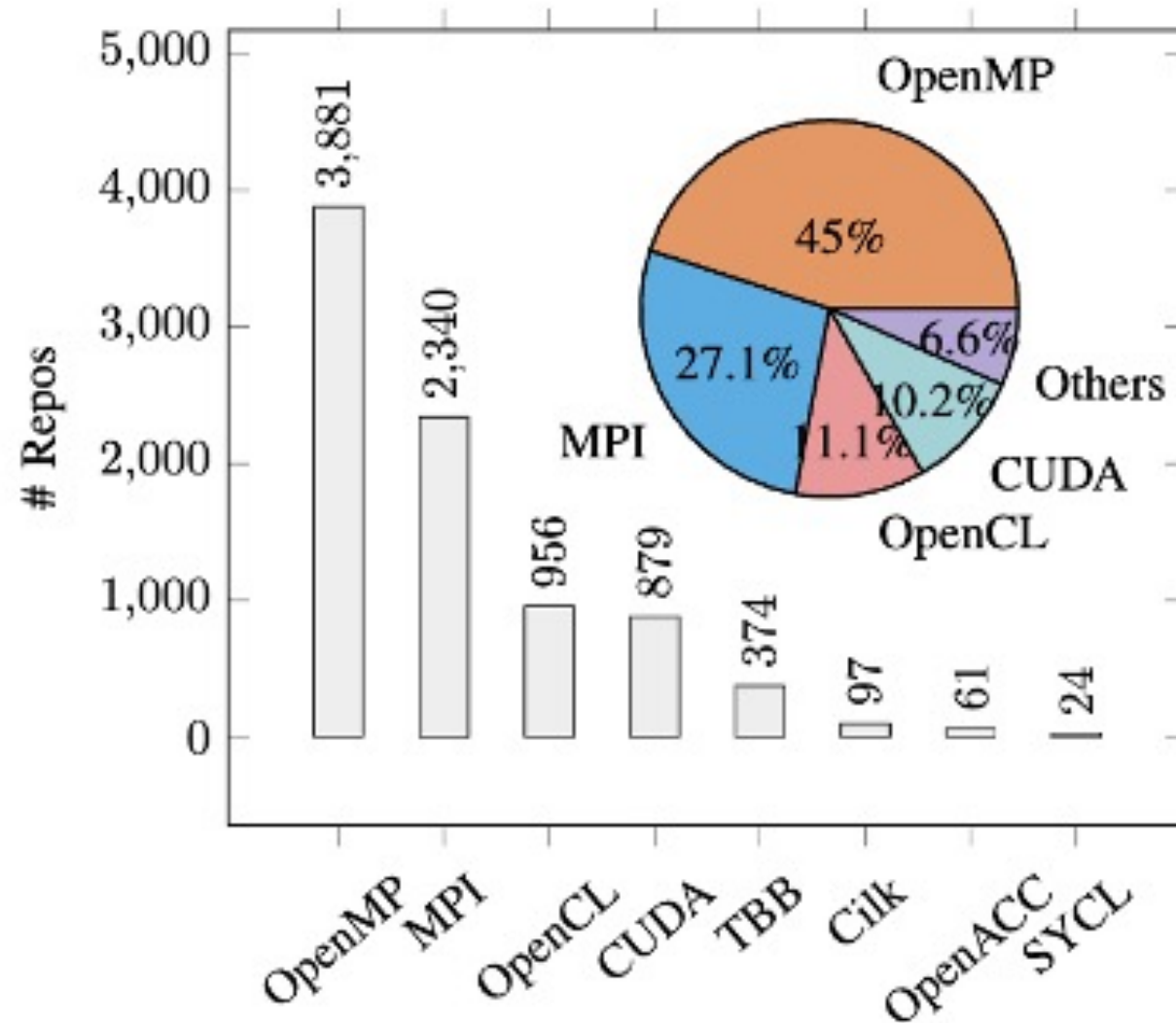
# Birth of OpenMP





# 25+ years later, OpenMP rules along side MPI

Parallel Programming model usage for C/C++/Fortran in publicly visible repositories in GitHub 2013 to 2023\*



We used a data-set created to train large language models for writing parallel code called HPCorpus\*.

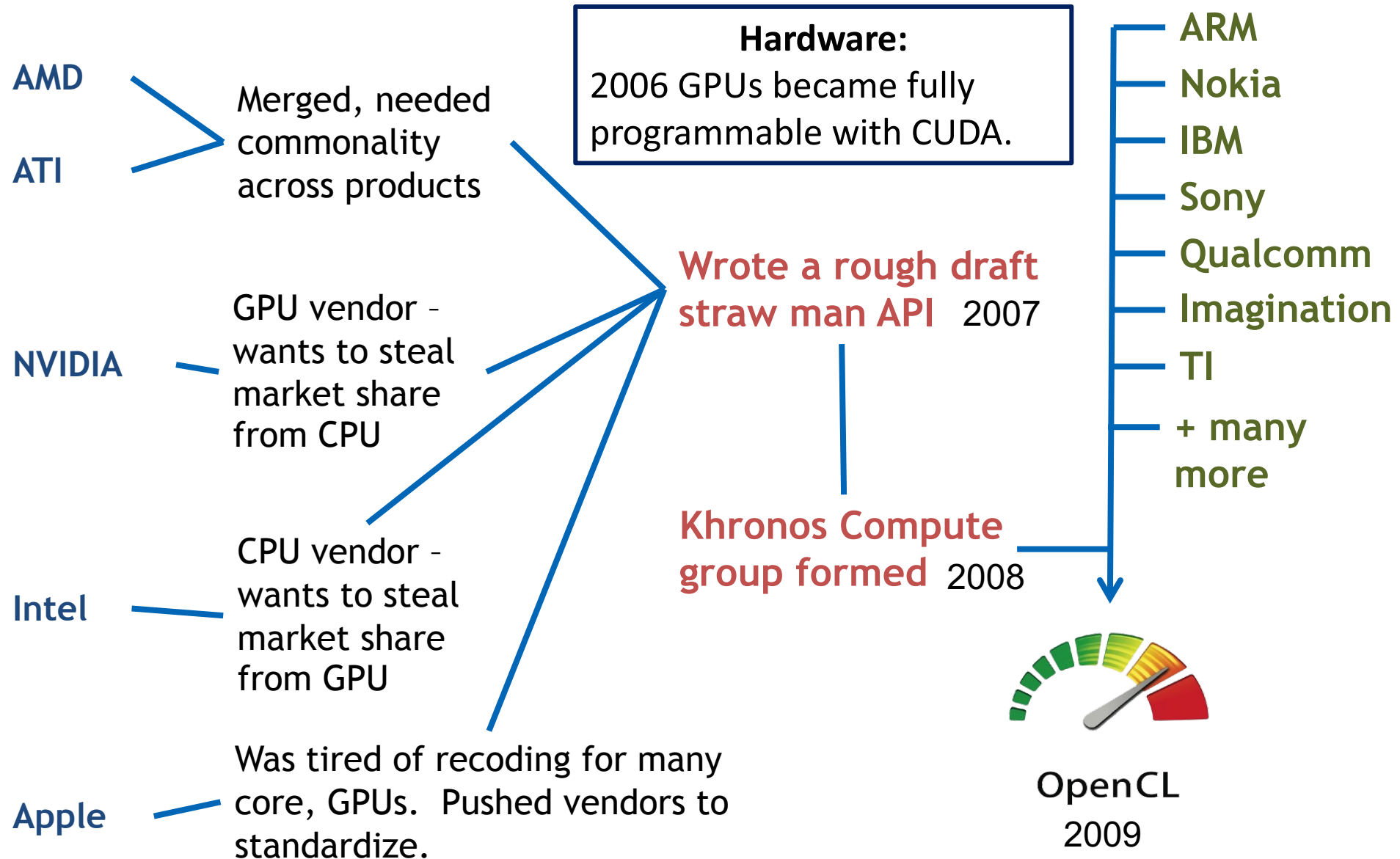
We scanned C, C++, and Fortran code inside visible repositories on Github.

We did not collect files with .cu or .cuf suffices, hence CUDA usage is undercounted.



# Birth of OpenCL

A partnership between vendors and application developers worked for clusters and SMPs ... surely it would work for GPUs?





# Birth of OpenCL

A partnership between vendors and application developers worked for clusters and SMPs ... surely it would work for GPUs?

AMD

ATI

Merged, needed  
commonality  
across products

## Hardware:

2006 GPUs became fully  
programmable with CUDA.

ARM

Nokia

IBM

Sony

Qualcomm

Wrote a rough draft

OpenCL for HPC failed ... leading to the mess we have today

- GPU programming is done with CUDA, OpenACC, OpenMP, Sycl, OpenCL, and more
- Application developers waste time figuring out which programming environment to use, when moving to new machines they may need to port their code, and vendors suffer as they need to support multiple options
- Given the money is in AI and not HPC ... vendors will not be much help solving this mess

group formed 2008

Intel

wants to steal  
market share  
from GPU

Apple

Was tired of recoding for many  
core, GPUs. Pushed vendors to  
standardize.



OpenCL  
2009



# We need OpenMP more than ever

- Less is more ... the fewer programming models “out there” the better. Application developers .... avoid new programming models unless external factors force a change! **You have more power than you think, but only if you pull together and ”speak with one voice”**

With so many parallel systems, people went wild and created numerous parallel programming environments ...

Parallel programming environments in the 90's

ABCPL	Cashmere	DICE	GLU	ISETL-Linda	Nano-Threads	Parafuse2	P-RIO	Split-C.
ACE	C4	DIPC	GUARD	ParLin	NESL	Paralation	Prospero	SR
ACT++	CC++	DOLIB	HASL	Eilean	NetClasses++	Parallel-C++	Proteus	Sthreads
Active messages	Chu	DDME	Haskell	P4-Linda	Nexus	Parallaxis	QPC++	Strand.
Ada	Charlotte	DOSMOS	HPC++	Glenda	Nimrod	ParC	PVM	SUIF
Adl	Charm	DRL	HPF	POSYBL	NOW	ParLib++	PSI	Synergy
Adsmith	Charm++	DSM-Threads	JAVAR.	Objective-Linda	Nx	ParLin	PSDM	Telegraphos
ADDDAP	Cid	Ease	HORUS	LIPS	Objective	Parmaes	Quake	SuperPascal
AFAP1	Clk	ECO	HPC	Locust	Linda	Pari	Quark	TCGMSG.
ALWAN	CM-Fortran	Eiffel	IMPACT	Lparx	Occam	pC	Quick	Threads.h++.
AM	Converse	Eilean	ISIS	Lucid	Omega	pC++	Threads	TreadMarks
AMDC	Code	Emerald	JAVAR	Maie	OpenMP	PCN	Sage++	TRAPPER
AppLeS	COOL	EPL	JADE	Manifold Mental	Oca	PCP	SCANDAL	uC++
Amoeba	CORRELATE	Excalibur	Java RMI	Legion	OOF90	PH	SAM pC++	UNITY
ARTS	CparPar	Express	javaPG	Meta Chaos	P++	PEACE	SCHEDULE	UC
Athapascan-0b	CPS	Falcon	JavaSpace	Midway	P3L	PCU	SciTL	V
Aurora	CRL	Filaments	JIDL	Millipede	P4	PET	POET	Vic*
Automap	CSP	FM	Joyce	Mirage	P4-Linda	PETSc	SDDA	Visifold V-NUS
bb_threads	Cthreads	FLASH	Khoros	MpC	Pablo	PENNY	SHMEM	VPE
Blaze	CUMULVS	The FORCE	Karma	MOSIX	PADE	Phosphorus	SIMPLE	Win32 threads
BSP	DAGGER	Fork	KOAN/Fortran-S	Modula-P	PADRE	POET	Sina	WinPar
BlockComm	DAPPLE	Fortran-M	LAM	Modula-2*	Panda	Polaris	SISAL	WWWinda
C*	Data Parallel C	FX	Lilac	Multipol	Papers	POOMA	distributed	XENOOOPS
C* in C	DC++	GA	Linda	MPI	AFAP1	POOL-T	smalltalk	XPC
C**	DCE++	GAMMA	JADA	MPC++	Para++	PRESTO	SMI	Zounds
CarlOS	DDD	Glenda	WWWinda	Munin	Paradigm		SONIC	ZPL



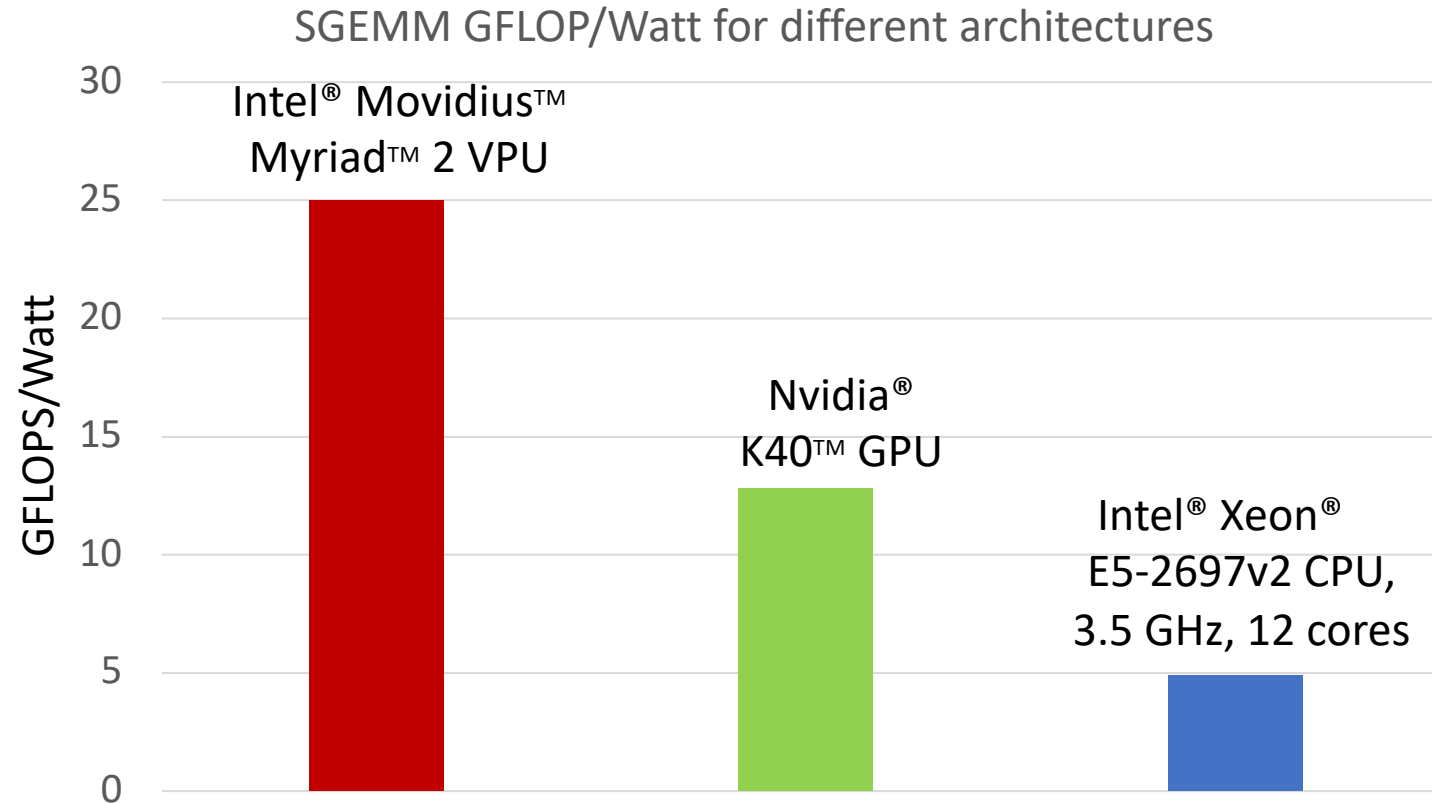
**In a world constrained by power ...  
computing will increasingly demand  
heterogeneous solutions**





# If you care about power, the world is heterogeneous?

Specialized processors doing operations suited to their architecture are more efficient than general purpose processors.

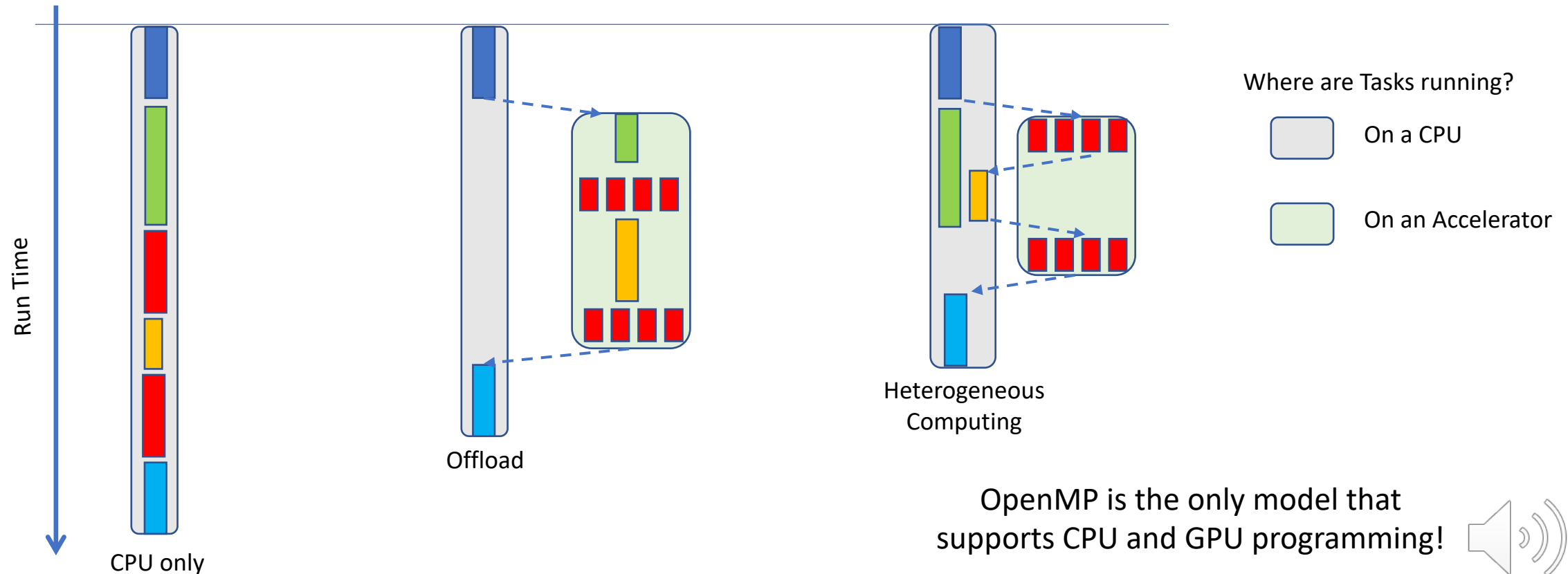


Hence, future systems will be increasingly heterogeneous ... GPUs, CPUs, FPGAs, and a wide range of accelerators



# Offload vs. Heterogeneous computing

- **Offload:** The CPU moves work to an accelerator and waits for the answer.
- **Heterogeneous Computing:** Run sub-problems in parallel on the hardware best suited to them.

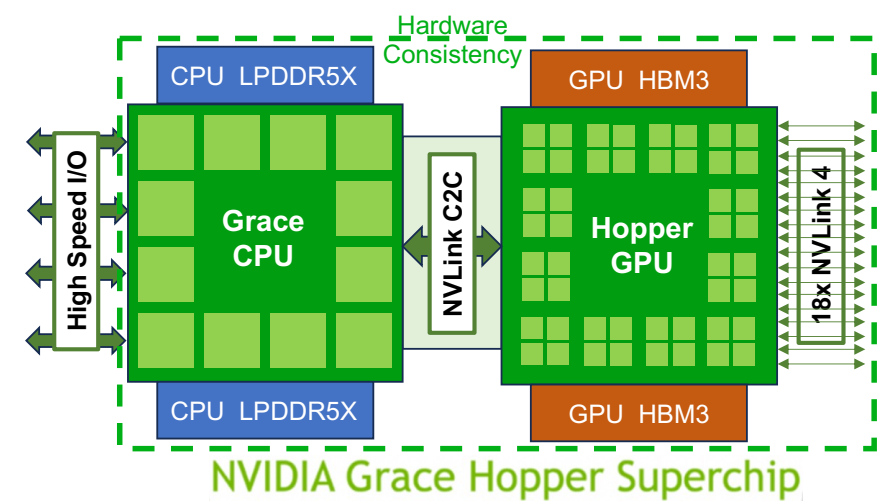


# Nvidia, AMD, and Intel are onboard

Nvidia released their heterogeneous "Grace Hopper superchip" in September 2022.

- 144 ARMv9 CPU cores and the Nvidia GH100 GPUs with 144 streaming multiprocessors and 576 fourth generation tensor cores
- Ongoing developments with Blackwell GPUs and future "Vera Rubin" CPU/GPUs.

<https://developer.nvidia.com/blog/nvidia-grace-cpu-superchip-architecture-in-depth/>

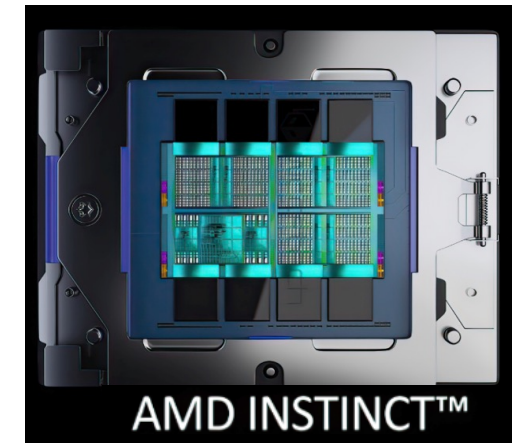


AMD INSTINCT™ products ... "World's first integrated data center CPU + GPU."

- MI300A: Zen 4 CPU + 8 CDNA-3 GPU chiplets deliver peak 61.3 FP64 TFLOPS
- Mi400 series planned for 2026 will be a major upgrade to the AMD INSTINCT™ product line.
- The Discovery supercomputer to be installed at ORNL in 2028 will use next-gen AMD EPYC CPUs, codenamed "Venice," and AMD Instinct™ MI430X GPUs.

<https://wccfttech.com/amd-confirms-next-gen-instinct-mi400-series-ai-accelerators-already-in-the-works/>

<https://www.ornl.gov/news/ornl-amd-and-hpe-deliver-does-newest-ai-supercomputers-discovery-and-lux>



Intel/Nvidia partnership announced Sept. 18, 2025: Intel x86 CPUs plus Nvidia GPUs and NVLink.

- "Intel x86 RTX SOC": A high-end client x86 CPU + Nvidia RTX GPU chiplet integrated through NVLink and packaged to fit in a single socket format. Uniform Memory Architecture
- Custom x86 Xeon™ CPUs integrated with Nvidia GPUs through NVLink. No details announced.



<https://nvidianews.nvidia.com/news/nvidia-and-intel-to-develop-ai-infrastructure-and-personal-computing-products>

<https://www.tomshardware.com/pc-components/cpus/nvidia-and-intel-announce-jointly-developed-intel-x86-rtx-socs-for-pcs-with-nvidia-graphics-also-custom-nvidia-data-center-x86-processors-nvidia-buys-usd5-billion-in-intel-stock-in-seismic-deal>

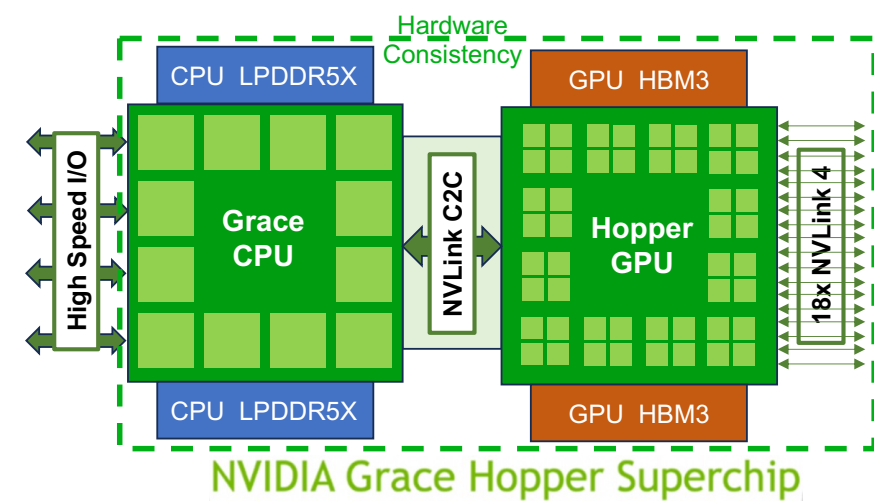


# Nvidia, AMD, and Intel are onboard

Nvidia released their heterogeneous "Grace Hopper superchip" in September 2022.

- 144 ARMv9 CPU cores and the Nvidia GH100 GPUs with 144 streaming multiprocessors and 576 fourth generation tensor cores
- Ongoing developments with Blackwell GPUs and future "Vera Rubin" CPU/GPUs.

<https://developer.nvidia.com/blog/nvidia-grace-cpu-superchip-architecture-in-depth/>



AMD INS  
• MI30  
• Mi40  
• The D  
coder

This means we need a portable, cross-vendor programming model that seamlessly integrates the latency sensitive parallelism of CPUs and the throughput optimized parallelism of GPUs.

<https://wccfte>  
<https://www.d>

The only portable option I know of today is OpenMP.

Intel/Nvidia partnership announced Sept. 18, 2025: Intel x86 CPUs plus Nvidia GPUs and NVLink.

- "Intel x86 RTX SOC": A high-end client x86 CPU + Nvidia RTX GPU chiplet integrated through NVLink and packaged to fit in a single socket format. Uniform Memory Architecture
- Custom x86 Xeon™ CPUs integrated with Nvidia GPUs through NVLink. No details announced.



<https://nvidianews.nvidia.com/news/nvidia-and-intel-to-develop-ai-infrastructure-and-personal-computing-products>

<https://www.tomshardware.com/pc-components/cpus/nvidia-and-intel-announce-jointly-developed-intel-x86-rtx-socs-for-pcs-with-nvidia-graphics-also-custom-nvidia-data-center-x86-processors-nvidia-buys-usd5-billion-in-intel-stock-in-seismic-deal>

# We need OpenMP more than ever

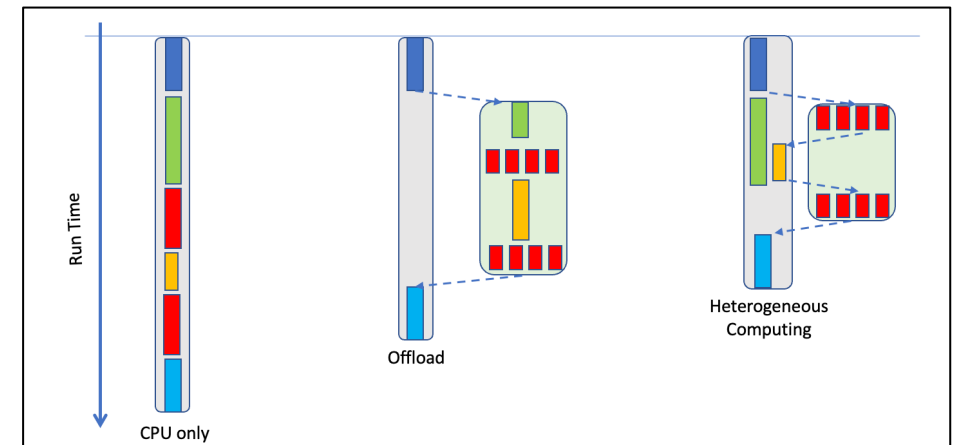
- Less is more ... the fewer programming models “out there” the better. Application developers .... avoid new programming models unless external factors force a change! **You have more power than you think, but only if you pull together and ”speak with one voice”**

With so many parallel systems, people went wild and created numerous parallel programming environments ...

Parallel programming environments in the 90's

ABCPCL	Cashmere	DICE	GLU	ISETL-Linda	Nano-Threads	Parafuse2	P-RIO	Split-C.
ACE	C4	DIPC	GUARD	ParLin	NESL	Paralation	Propero	SR
ACT++	CC++	DOLIB	HASL	Eilean	NetClasses++	Parallel-C++	Proteus	Shtreads
Active messages	Chu	DOME	Haskell	P4-Linda	Nexus	Parallaxis	QPC++	Strand.
Ada	Charlotte	DOSMOS	HPC++	Glenda	Nimrod	ParC	SUIF	PVM
Adl	Charm	DRL	HPF	POSYBL	NOW	ParLib++	PSI	Synergy
Adsmith	Charm++	DSM-Threads	JAVAR	Objective-Linda	Nx	ParLin	PSDM	Teleghos
ADDDAP	Cid	Ease	HORUS	LIPS	Objective	Parmaes	Quake	SuperPascal
AFAP1	Clk	ECO	HPC	Locust	Linda	Pari	Quark	TCOMSG
ALWAN	CM-Fortran	Eiffel	IMPACT	Lparx	Occam	pC	Quick	Threads.h++
AM	Converse	Eilean	ISIS	Lucid	Omega	pC++	Threads	TreadMarks
AMDC	Code	Emerald	JAVAR	Maiee	OpenMP	PCN	Sage++	TRAPPER
AppLeS	COOL	EPL	JADE	Manifold Mental	Oca	PCP	SCANDAL	uC++
Amoeba	CORRELATE	Excalibur	Java RMI	Legion	OOF90	PH	SAM pC++	UNITY
ARTS	CparPar	Express	javaPG	Meta Chaos	P++	PEACE	SCHEDULE	UC
Athapascan-0b	CPS	Falcon	JavaSpace	Midway	P3L	PCU	ScTL	V
Aurora	CRL	Filaments	JIDL	Millipede	P4	PET	POET	Vic*
Automap	CSP	FM	Joyce	Mirage	P4-Linda	PETSc	SDDA	Visifold V-NUS
bb_threads	Cthreads	FLASH	Khoros	MpC	Pablo	PENNY	SHMEM	VPE
Blaze	CUMULVS	The FORCE	Karma	MOSIX	PADE	Phosphorus	SIMPLE	Win32 threads
BSP	DAGGER	Fork	KOAN/Fortran-S	Module-P	PADRE	POET	Sina	WinPar
BlockComm	DAPPLE	Fortran-M	LAM	Module-2*	Panda	Polaris	SISAL	WWWinda
C*	Data Parallel C	FX	Lilac	Multipol	Papers	POOMA	distributed	XENOOOPS
C* in C	DC++	GA	Linda	MPI	AFAP1	POOL-T	smalltalk	XPC
C**	DCE++	GAMMA	JADA	MPC++	Para++	PRESTO	SONIC	ZPL
CarlOS	DDD	Glenda	WWWinda	Munin	Paradigm			

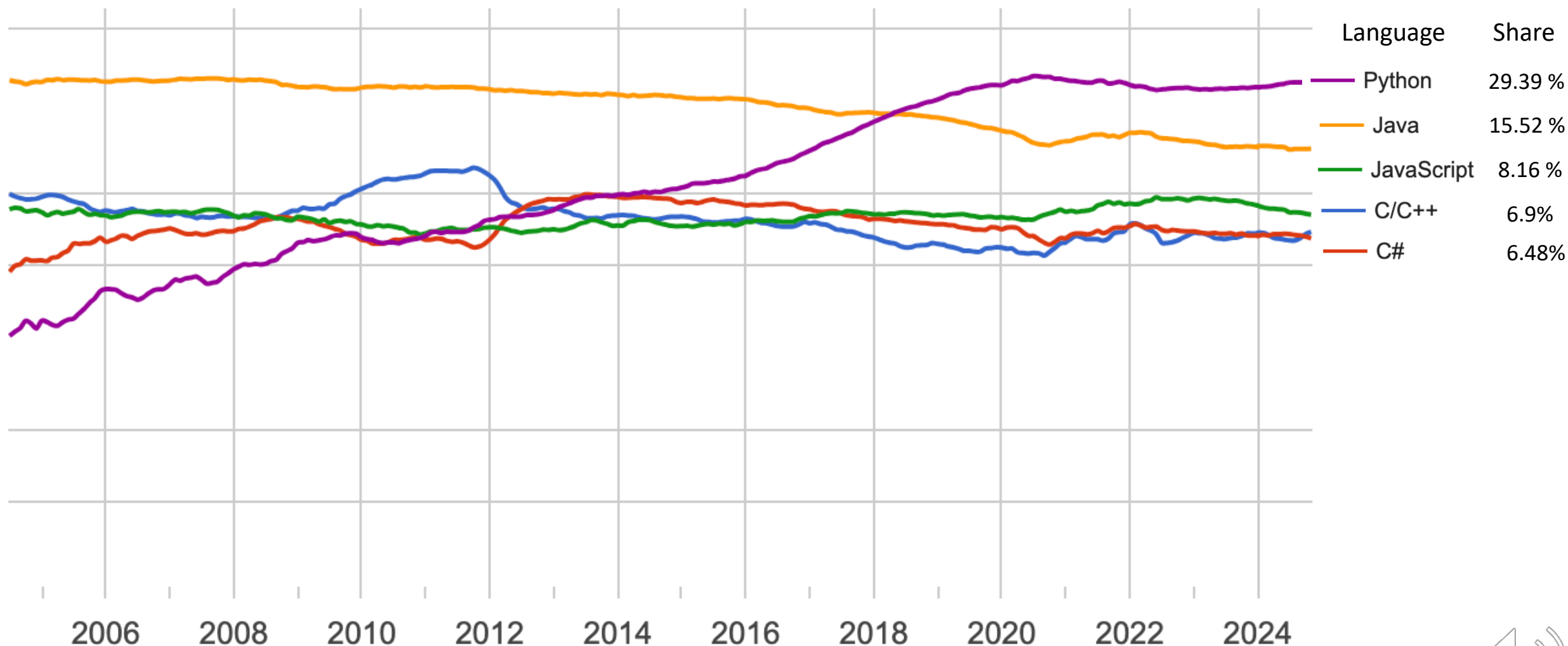
- Heterogeneous computing ... our programs must exploit the power/performance benefits of mapping each part of an application to the processor that runs it best. **OpenMP is one of the only portable/performant programming models that handles CPUs and GPUs**



**There never has been and never will be  
one programming language to rule them all**

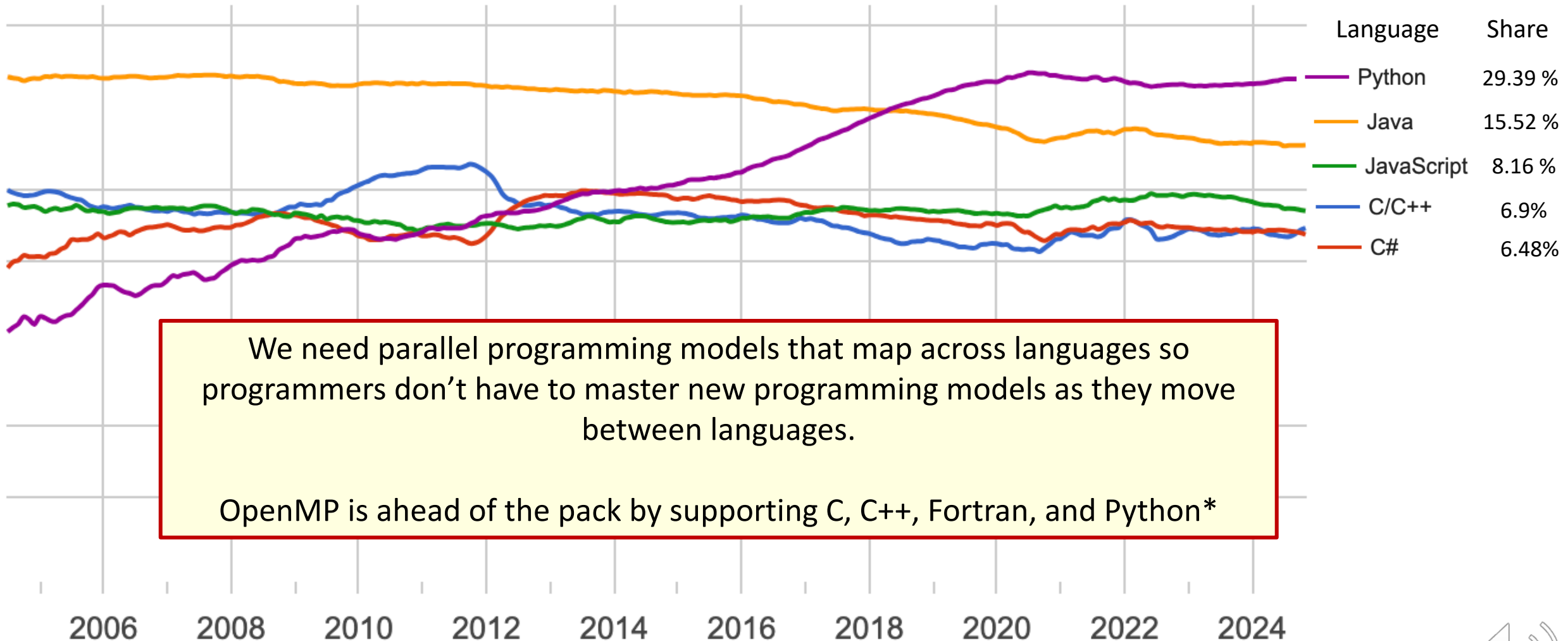


# Popularity of Programming Languages (PYPL)





# Popularity of Programming Languages (PYPL)



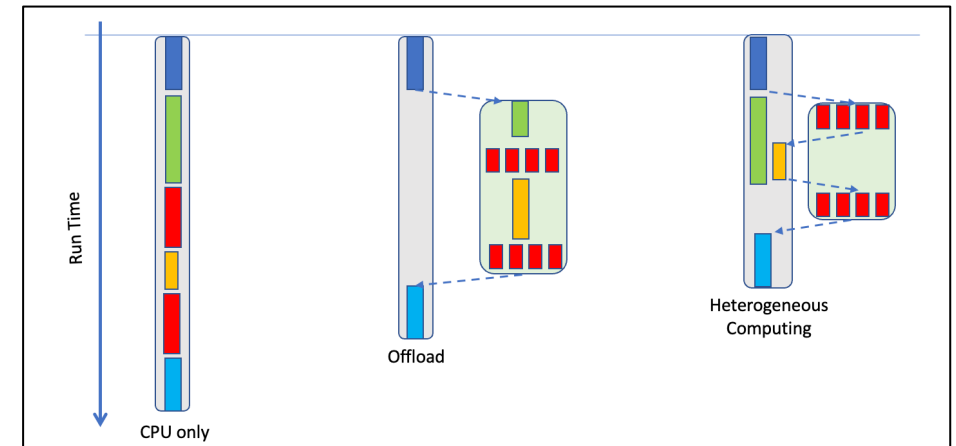
# We need OpenMP more than ever

- Less is more ... the fewer programming models “out there” the better. Application developers .... avoid new programming models unless external factors force a change! **You have more power than you think, but only if you pull together and “speak with one voice”**

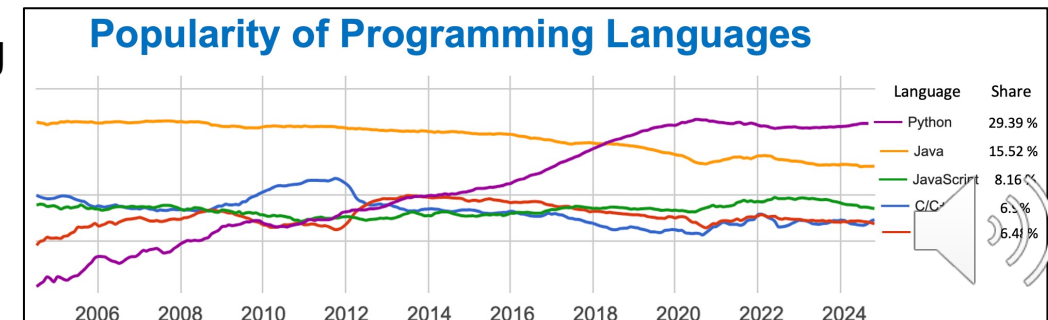
With so many parallel systems, people went wild and created numerous parallel programming environments ...

ABCPCL	Cashmere	DICE	GLU	ISLTL-Linda	Nano-Threads	Parafuse2	P-RIO	Split-C.
ACE	C4	DIPC	GUARD	ParLin	NESL	Paralation	Prospero	SR
ACT++	CC++	DOLIB	HASL	Eilean	NetClasses++	Parallel-C++	Proteus	Sthreads
Active messages	Chu	DOME	Haskell	P4-Linda	Nexus	Parallaxis	QPC++	Strand.
Ada	Charlotte	DOSMOS	HPC++	Glenda	Nimrod	ParC	SUIF	SUIF
Adl	Charm	DRL	HPF	POSYBL	NOW	ParLib++	PSI	Synergy
Adsmith	Charm++	DSM-Threads	JAVAR	Objective-Linda	Nx	ParLin	PSDM	Telegraphos
ADDDAP	Cid	Ease	HERUS	LIPS	Objective	Parma	Quake	SuperPascal
AFAP	Clk	ECO	HPC	Locust	Linda	Pari	Quark	TCOMGG
ALWAN	CM-Fortran	Eiffel	IMPACT	Lparx	Occam	pC	Quick	Threads.h++
AM	Converse	Eilean	ISIS	Lucid	Omega	pC++	Threads	TreadMarks
AMDC	Code	Emerald	JAVAR	Maie	OpenMP	PCN	Sage++	TRAPPER
AppLeS	COOL	EPL	JADE	Manifold Mentat	Oca	PCP	SCANDAL	UNITY
Amoeba	CORRELATE	Excelsior	Java RMI	Legion	OOF90	PH	SAM pC++	UC
ARTS	CparPar	Express	javaPG	Meta Chaos	P++	PEACE	SCHEDULE	V
Athapascan-0b	CPS	Falcon	JavaSpace	Midway	P3L	PCU	ScTL	Vicfold V-NUS
Aurora	CRL	Filaments	JIDL	Millipede	P4	PET	POET	VPE
Automap	CSP	FM	Joyce	Mirage	P4-Linda	PETSc	SDDA	Win32 threads
bb_threads	Cthreads	FLASH	Khoros	MpC	Pablo	PENNY	SHMEM	WinPar
Blaze	CUMULVS	The FORCE	Karma	MOSIX	PADE	Phosphorus	SIMPLE	XENOOOPS
BSP	DAGGER	Fork	KOAN/Fortran-S	Modula-P	PADRE	POET	Sina	XPC
BlockComm	DAPPLE	Fortran-M	LAM	Modula-2*	Panda	Polaris	SISAL	ZPL
C*	Data Parallel C	FX	Lilac	Multipol	Papers	POOMA	distributed	
C* in C	DC++	GA	Linda	MPI	AFAP	POOL-T	smalltalk	
C**	DCE++	GAMMA	JADA	MPC++	Para++	PRESTO	SONIC	
CarlOS	DDD	Glenda	WWWinda	Munin	Paradigm			

- Heterogeneous computing ... our programs must exploit the power/performance benefits of mapping each part of an application to the processor that runs it best. **OpenMP is one of the only portable/performant programming models that handles CPUs and GPUs**

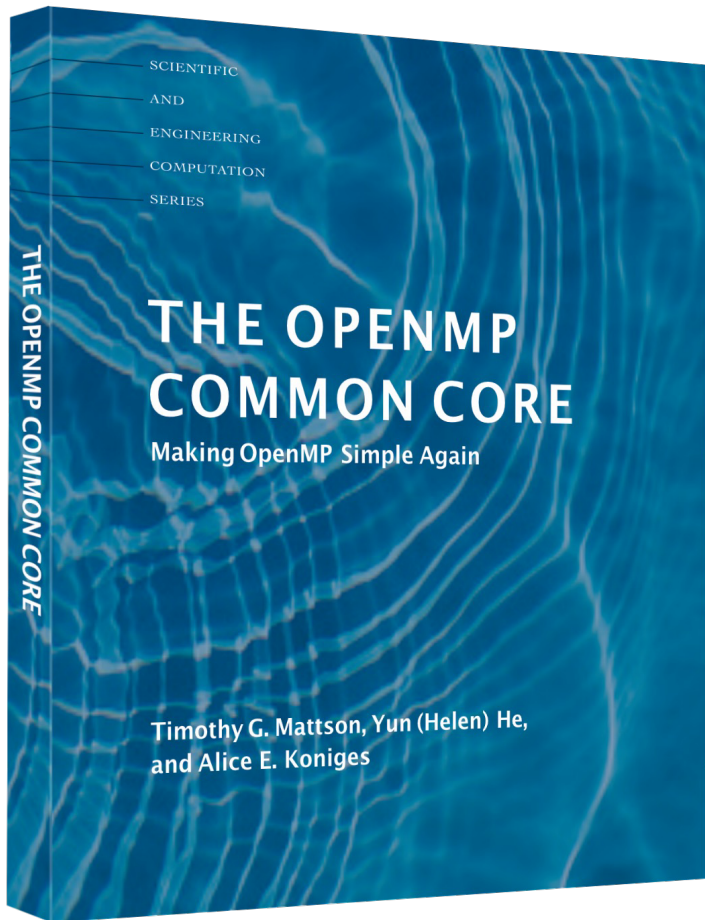


- No single language will rule the world ... parallel programming models that move between languages let's you exploit your parallel programming skills across languages. **We need OpenMP with C, C++, Fortran, and Python.**



# We need OpenMP more than ever

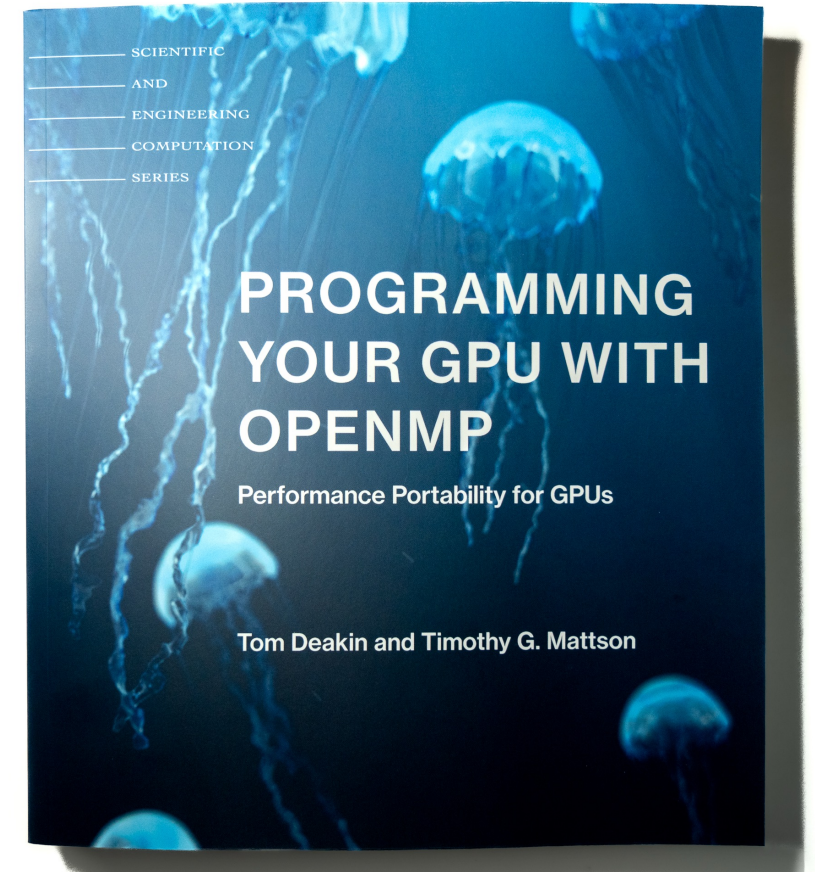
- And if you are new to OpenMP or just want to learn it more deeply, there are some great books to help you out



Start Here ...



Go beyond basic multithreading ...



Learn how to write portable code for GPUs.



# OpenMP<sup>®</sup>

## SC25 OpenMP Tech Talk Series



Why we need OpenMP  
more than ever

Tim Mattson | Human Learning Group