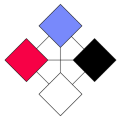


DAPHNE

https://daphne-eu.eu/

DaphneSched: A Scheduler for Integrated Data Analysis Pipelines

Jonas Müller Korndörfer, Quentin Guilloteau, Florina Ciorba

Scheduling for Large-scale Systems
Aussois, June 26, 2024

University of Basel

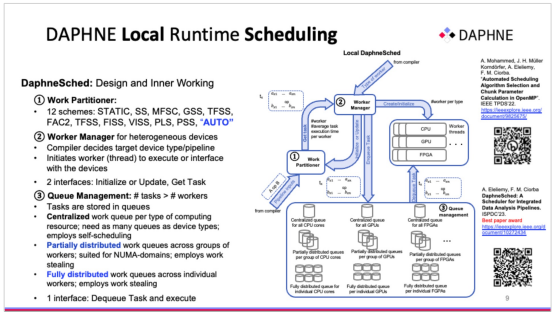
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 957407.

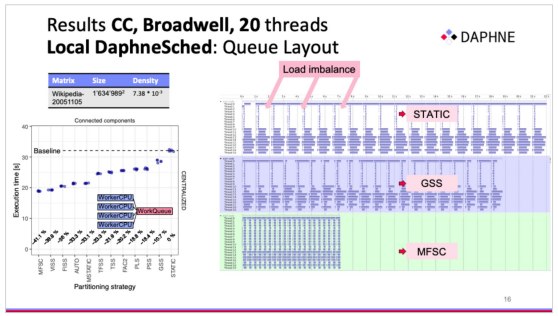
DMI HPC

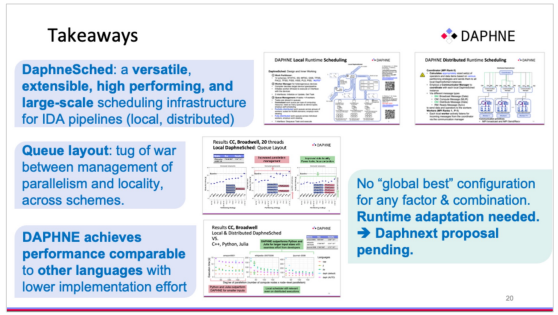
https://hpc.dmi.unibas.ch/

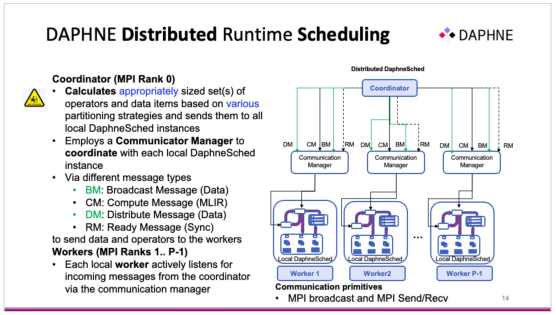
Overview

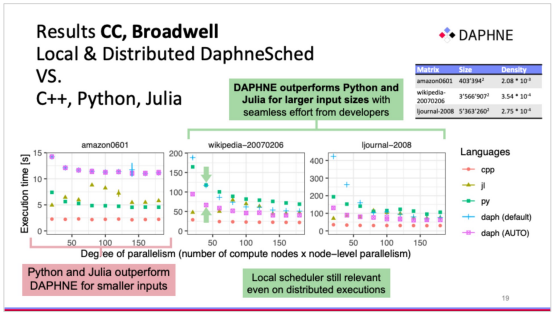
DAPHNE





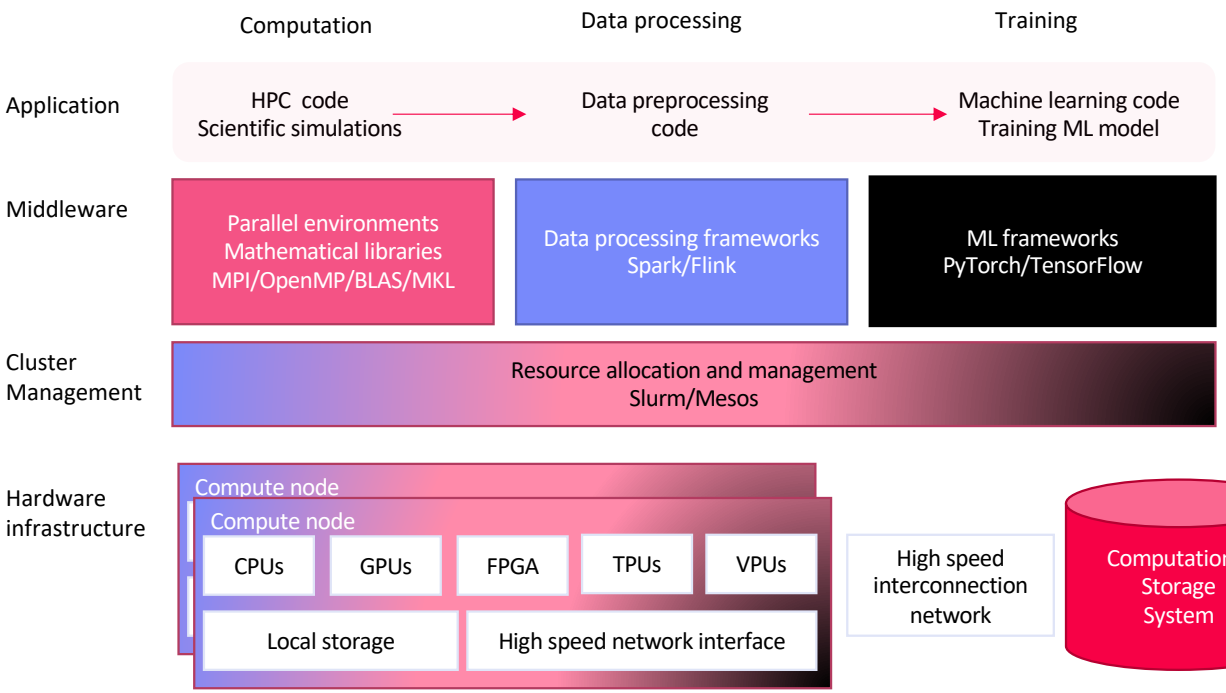






Motivation & Challenges (1/5)

DAPHNE



N. Ihde et al.
A Survey of Big Data, HPC and Machine Learning Benchmarks,
TPCTC 2021, Copenhagen, Denmark, August 2021
Open Access here

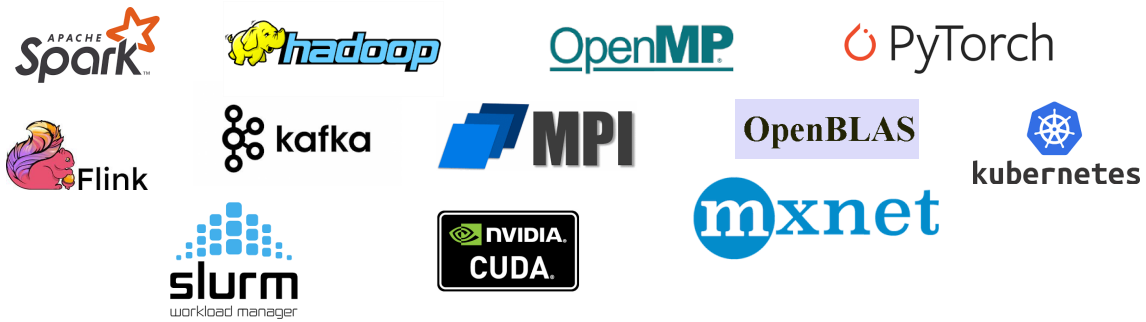
- Integrated data analysis (IDA) pipelines comprise data management, query processing, high performance computing, complex simulations, training and scoring for multiple machine learning models.
- IDA become increasingly common in practice, share compilation, runtime techniques, and converging cluster hardware

Motivation & Challenges (2/5)

DAPHNE

Different

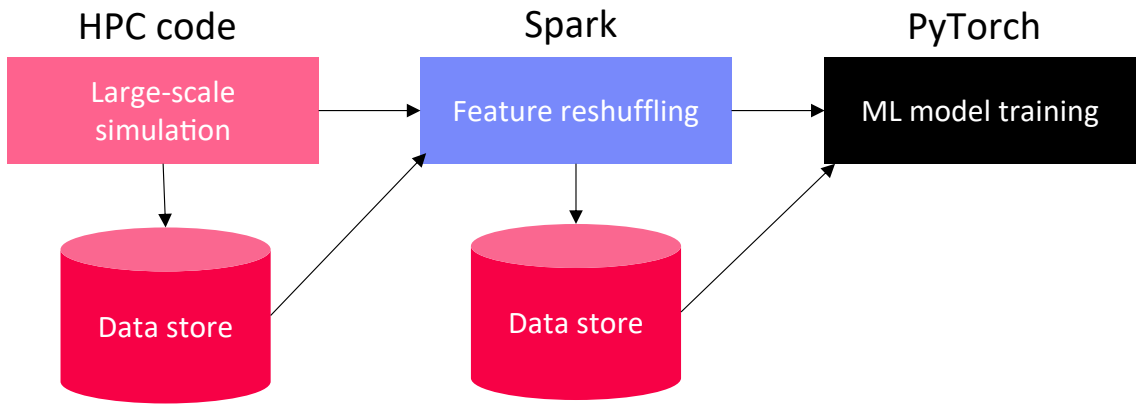
- system libraries
- programming models



Motivation & Challenges (3/5)

DAPHNE

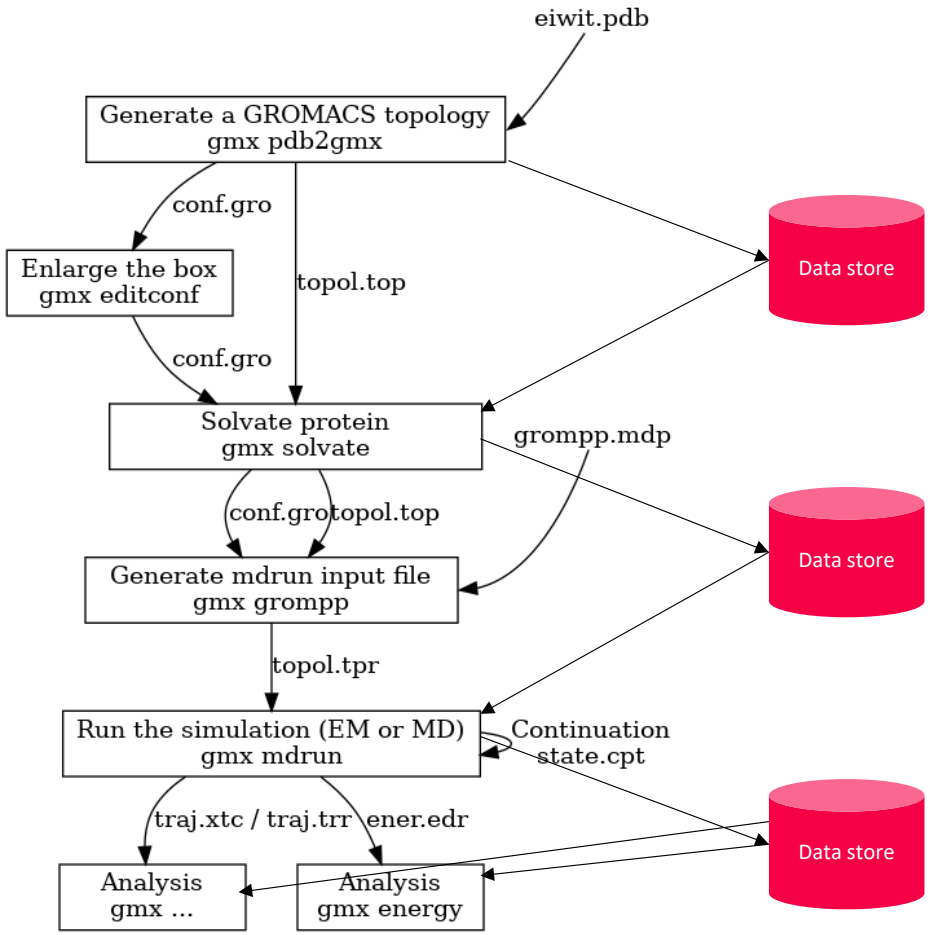
- Data exchange between various components is a disaggregated pipeline



Motivation & Challenges (4/5)

DAPHNE

- Data exchange between various components is a disaggregated pipeline
- Example: Typical GROMACS MD run of a protein in a box of water



Source: https://manual.gromacs.org/current/user-guide/flow.html

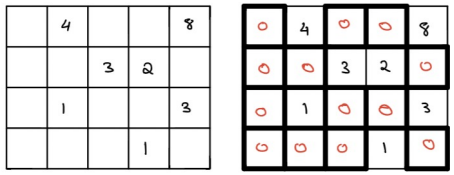
Motivation & Challenges (5/5)

DAPHNE

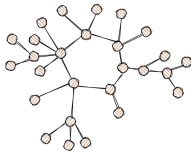
Various

- data types
- data representations
- input sizes

Dense vs Sparse



Graph



FP32, FP64, INT8,
INT32, INT64, UINT8,
BF16, TF32, FlexPoint

Overview of the DAPHNE System

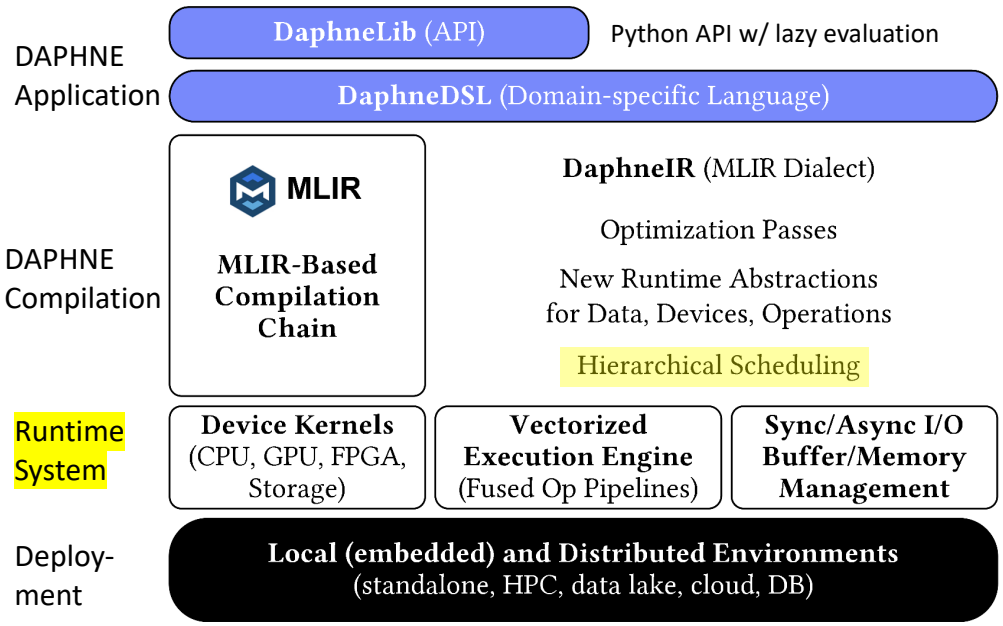
DAPHNE

"An Open and Extensible System Infrastructure for IDA Pipelines"

Patrick Damme et al.: **DAPHNE:** An Open and Extensible System Infrastructure for Integrated Data Analysis Pipelines [**CIDR 2022**] Open Access here

Daphne GitHub repository: here

System Architecture



Distributed and local vectorized execution

- Coarse grained tasks and cache-conscious data binding
- Fused operator pipelines on tiles/vectors of data
- Device kernels for heterogeneous hardware
- Integration of computational storage (e.g., eBPF programs)

DaphneSched

- Scheduling for load balancing (e.g., for ops on sparse data)

Daphne Runtime

- Different distributed backends (e.g., gRPC, OpenMPI)

DAPHNE **Local** Runtime **Scheduling**

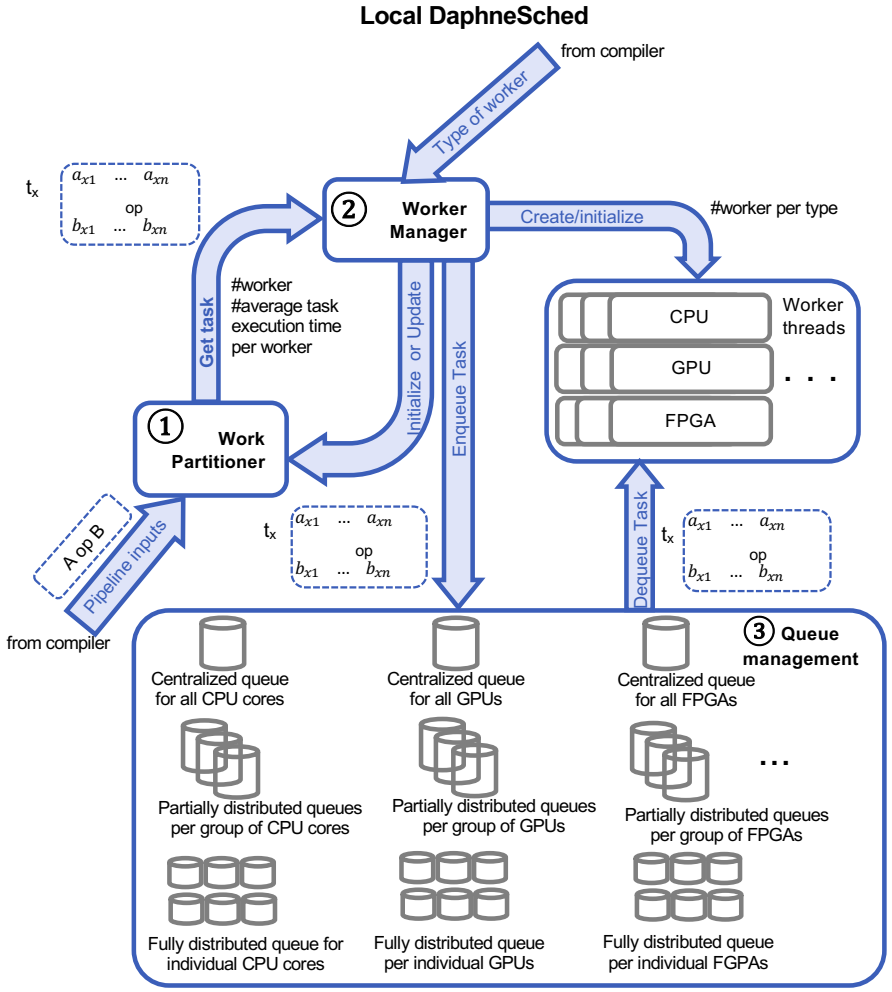
DAPHNE

DaphneSched: Design and Inner Working

① **Work Partitioner:**
- 12 schemes: STATIC, SS, MFSC, GSS, TFSS, FAC2, TFSS, FISS, VISS, PLS, PSS, **"AUTO"**

② **Worker Manager** for heterogeneous devices
- Compiler decides target device type/pipeline
- Initiates worker (thread) to execute or interface with the devices
- 2 interfaces: Initialize or Update, Get Task

③ **Queue Management:** # tasks > # workers
- Tasks are stored in queues
- **Centralized** work queue per type of computing resource; need as many queues as device types; employs self-scheduling
- **Partially distributed** work queues across groups of workers; suited for NUMA-domains; employs work stealing
- **Fully distributed** work queues across individual workers; employs work stealing
- 1 interface: Dequeue Task and execute



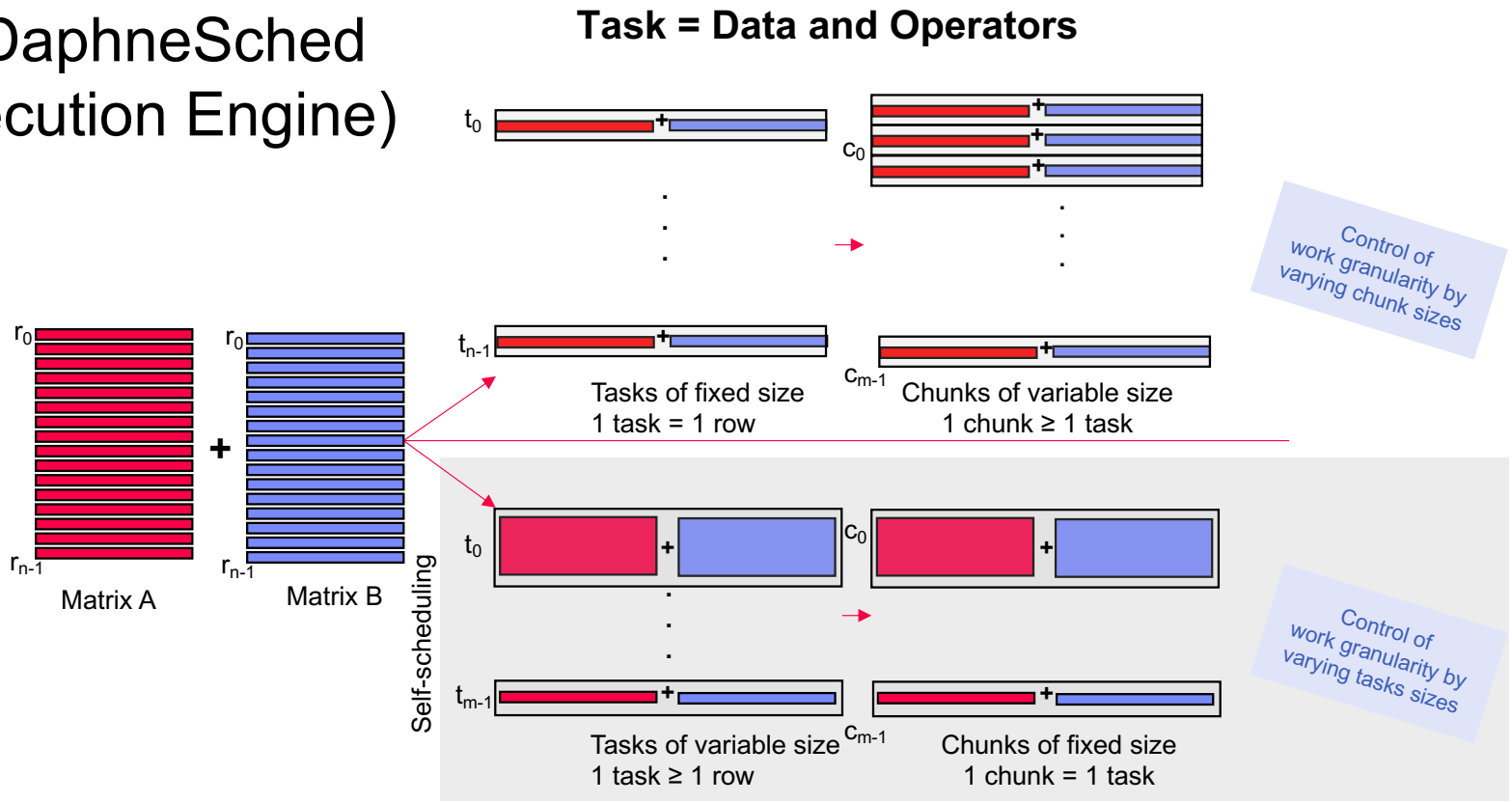
A. Mohammed, J. H. Müller Korndörfer, A. Eleliemy, F. M. Ciorba.
"**Automated Scheduling Algorithm Selection and Chunk Parameter Calculation in OpenMP**". IEEE TPDS'22.
https://ieeexplore.ieee.org/document/9825675/

A. Eleliemy, F. M. Ciorba
DaphneSched: A Scheduler for Integrated Data Analysis Pipelines. ISPDC'23.
Best paper award
https://ieeexplore.ieee.org/document/10272434

DAPHNE **Local** Runtime **Scheduling**

DAPHNE

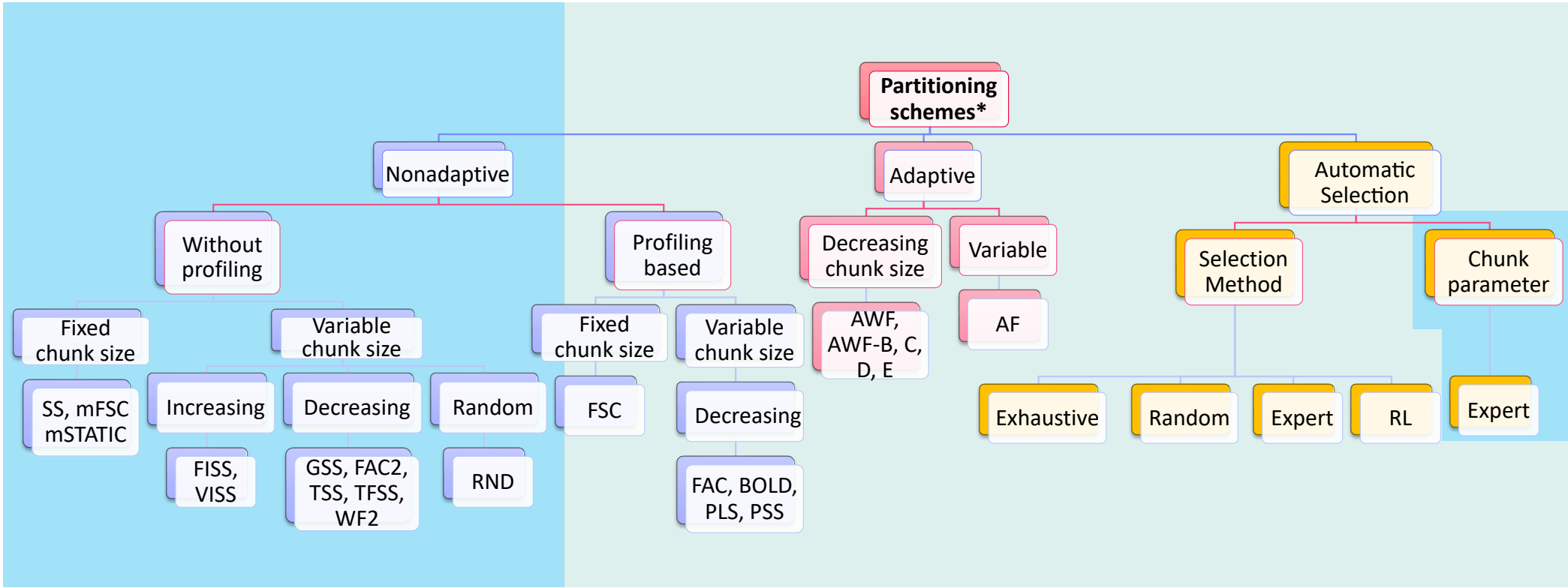
Task formation in DaphneSched
(by Vectorized Execution Engine)



DAPHNE **Local** Runtime **Scheduling**

DAPHNE

Work **Partitioning** Schemes



* Work partitioning employs the chunk calculation formulae of the various DLS schemes

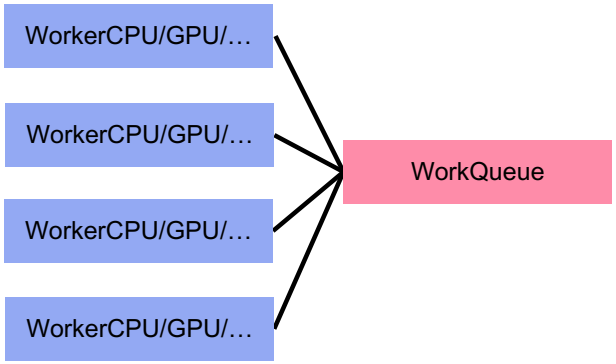
DAPHNE **Local** Runtime **Scheduling**

DAPHNE

Work **Queues** Organization

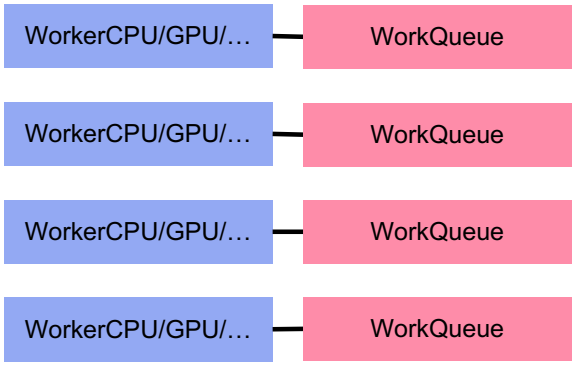
- Employs Work-sharing to fill queues

Centralized

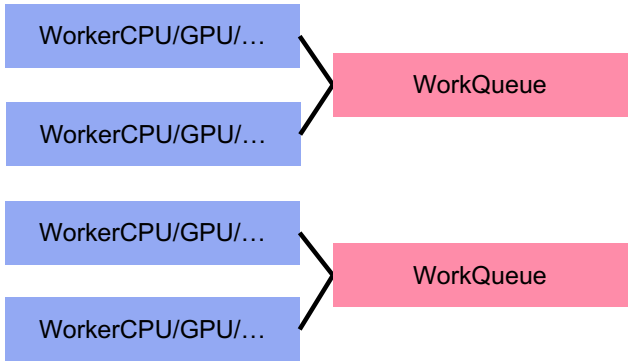


Employs: self-scheduling assignment

Per Device



Per Device Group



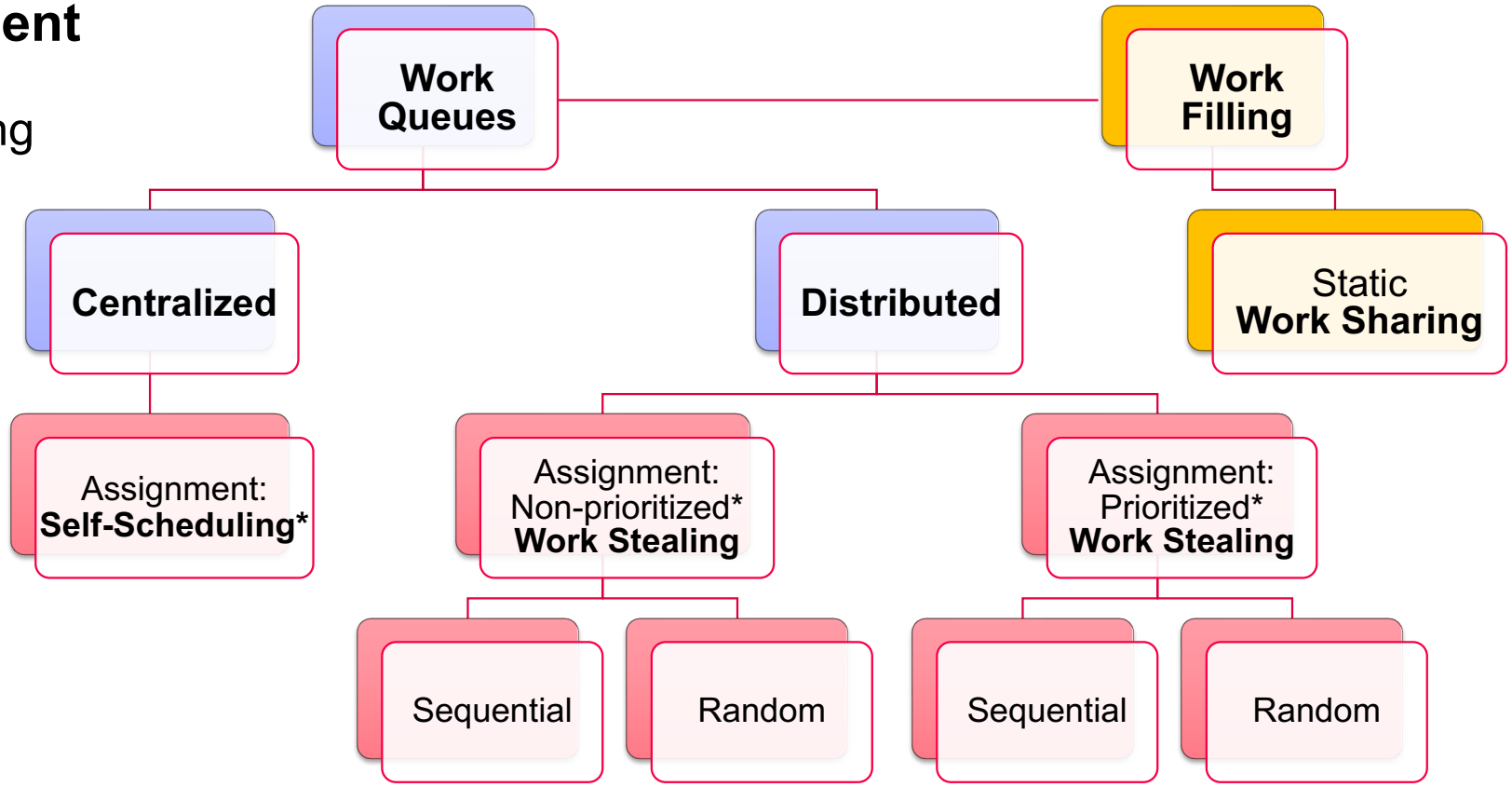
Employs: work stealing assignment

DAPHNE **Local** Runtime **Scheduling**

DAPHNE

Work **Assignment**

- Work sharing
- Self-scheduling
- Work stealing



* Prioritization: here device locality

DAPHNE **Distributed** Runtime **Scheduling**

DAPHNE

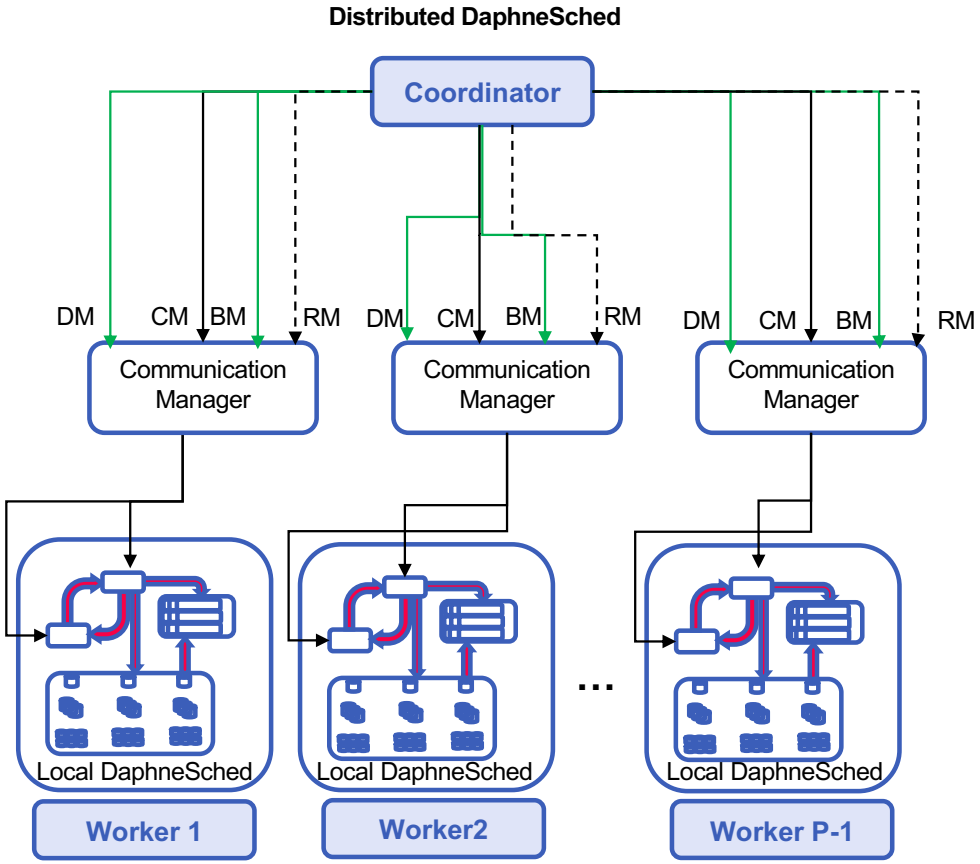
Coordinator (MPI Rank 0)

- **Calculates** appropriately sized set(s) of operators and data items based on various partitioning strategies and sends them to all local DaphneSched instances
- Employs a **Communicator Manager** to **coordinate** with each local DaphneSched instance
- Via different message types
  - BM: Broadcast Message (Data)
  - CM: Compute Message (MLIR)
  - DM: Distribute Message (Data)
  - RM: Ready Message (Sync)

to send data and operators to the workers

Workers (MPI Ranks 1.. P-1)

- Each local **worker** actively listens for incoming messages from the coordinator via the communication manager

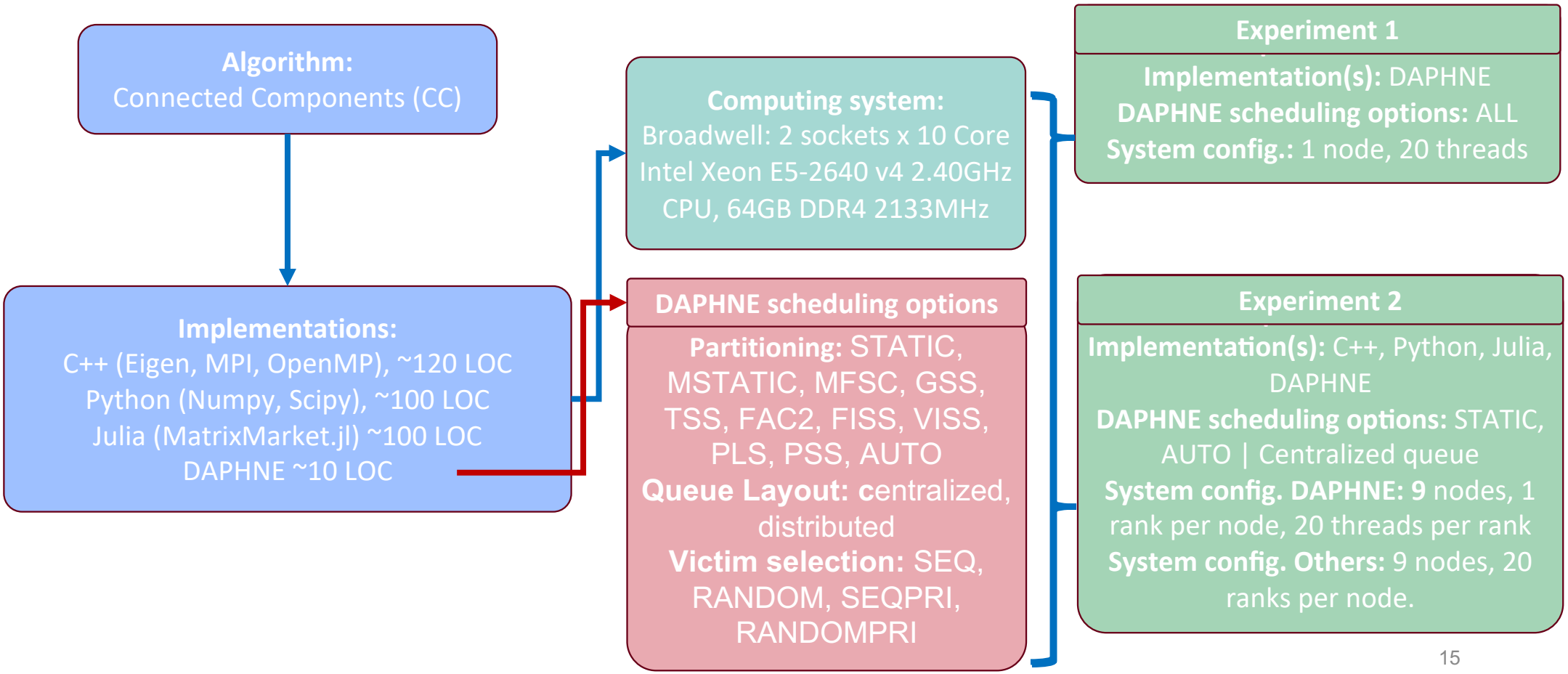


Communication primitives

- MPI broadcast and MPI Send/Recv

DaphneSched: Factorial Experiments

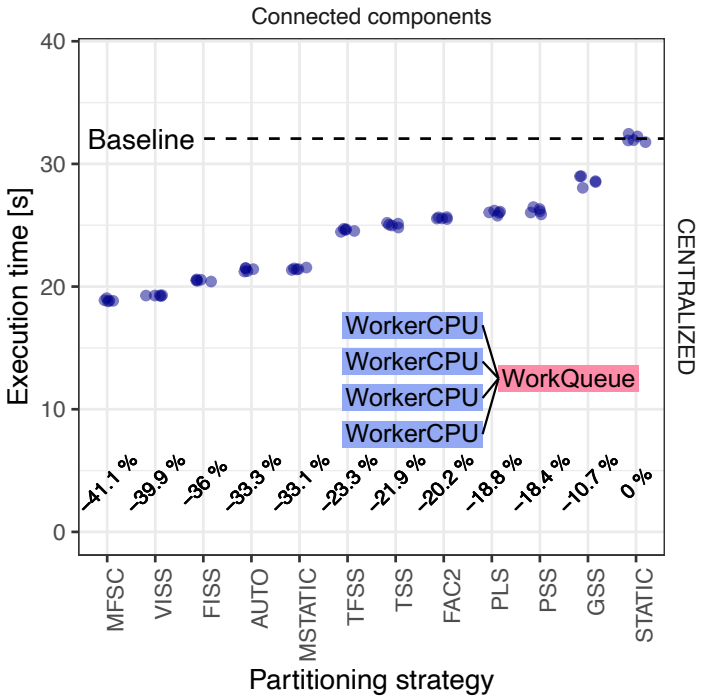
DAPHNE

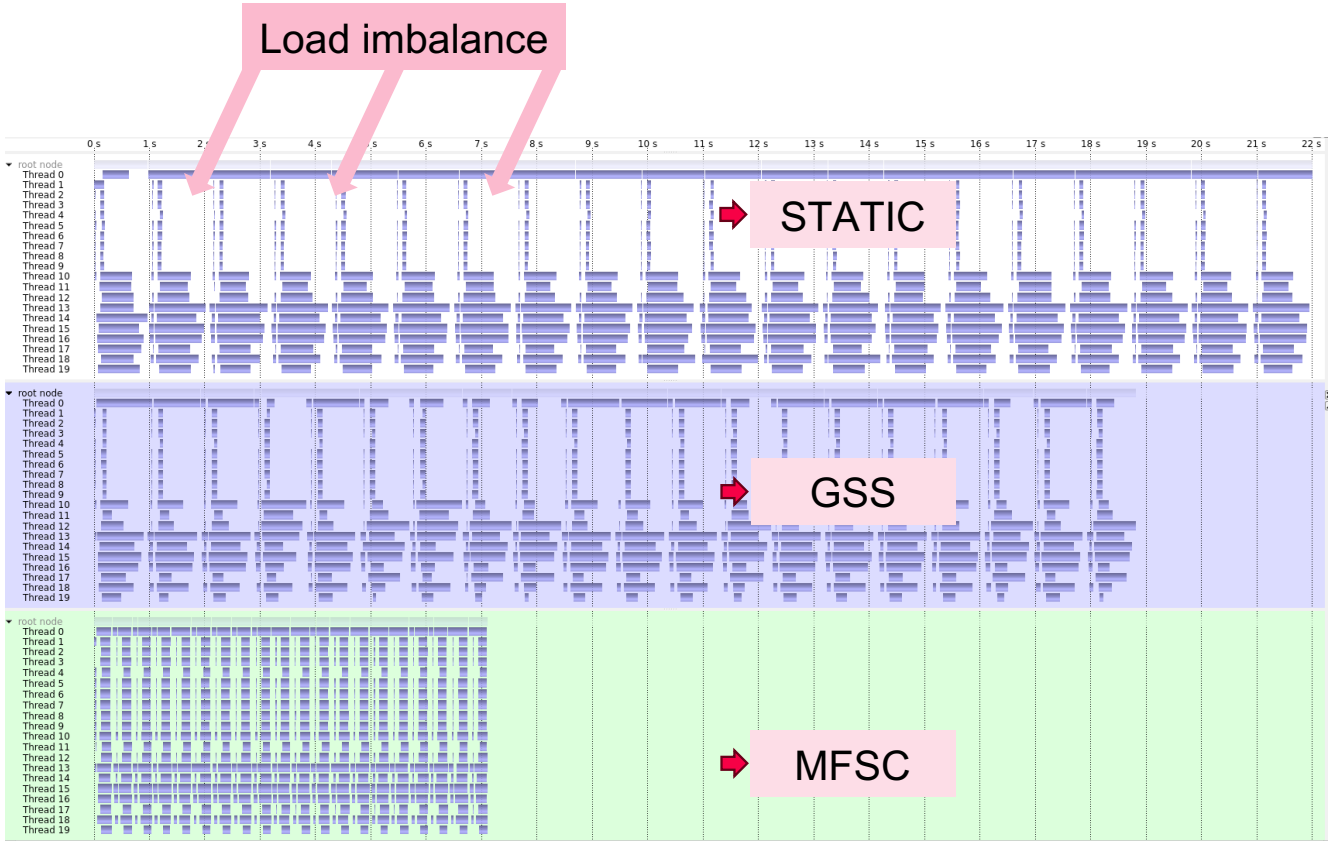


Results **CC, Broadwell, 20** threads
Local DaphneSched: Queue Layout

DAPHNE

Matrix	Size	Density
Wikipedia-20051105	$1'634'989^2$	$7.38 * 10^{-3}$



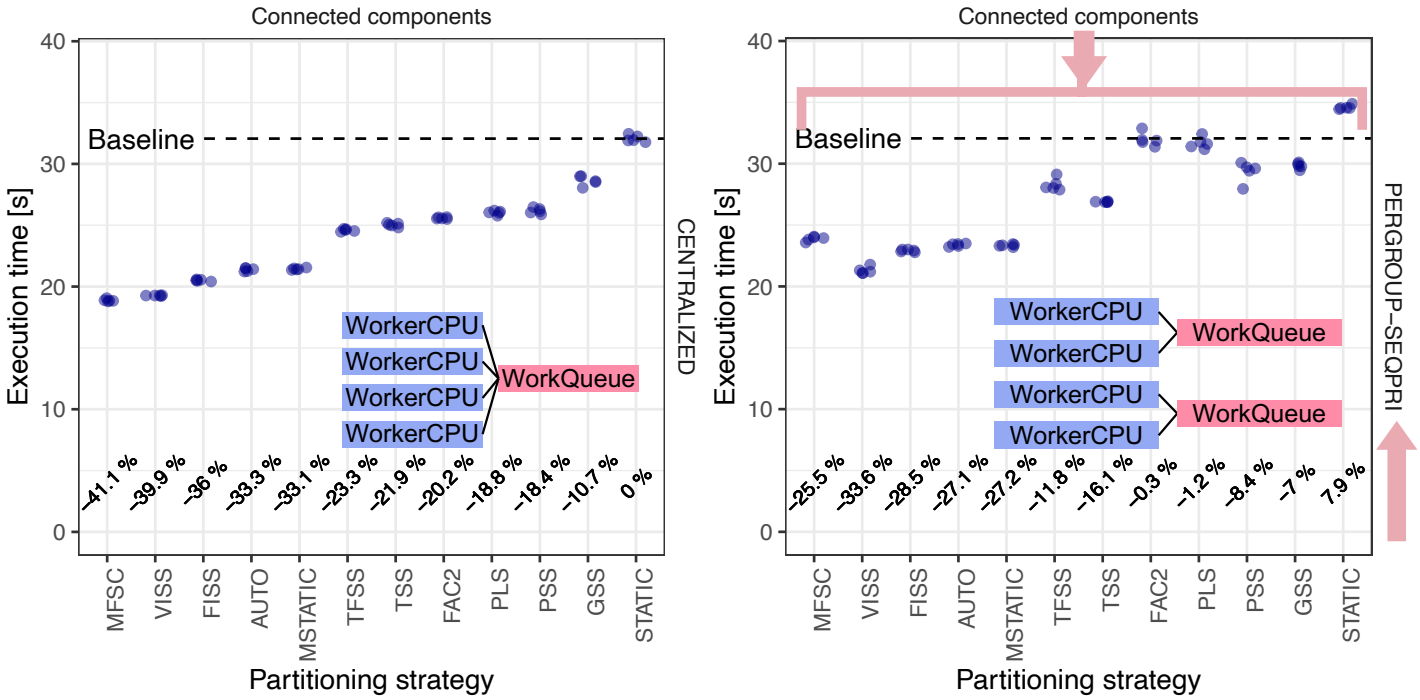


Results **CC, Broadwell, 20** threads
Local DaphneSched: Queue Layout

DAPHNE

Matrix	Size	Density
Wikipedia-20051105	$1'634'989^2$	$7.38 * 10^{-3}$

Increased parallelism management



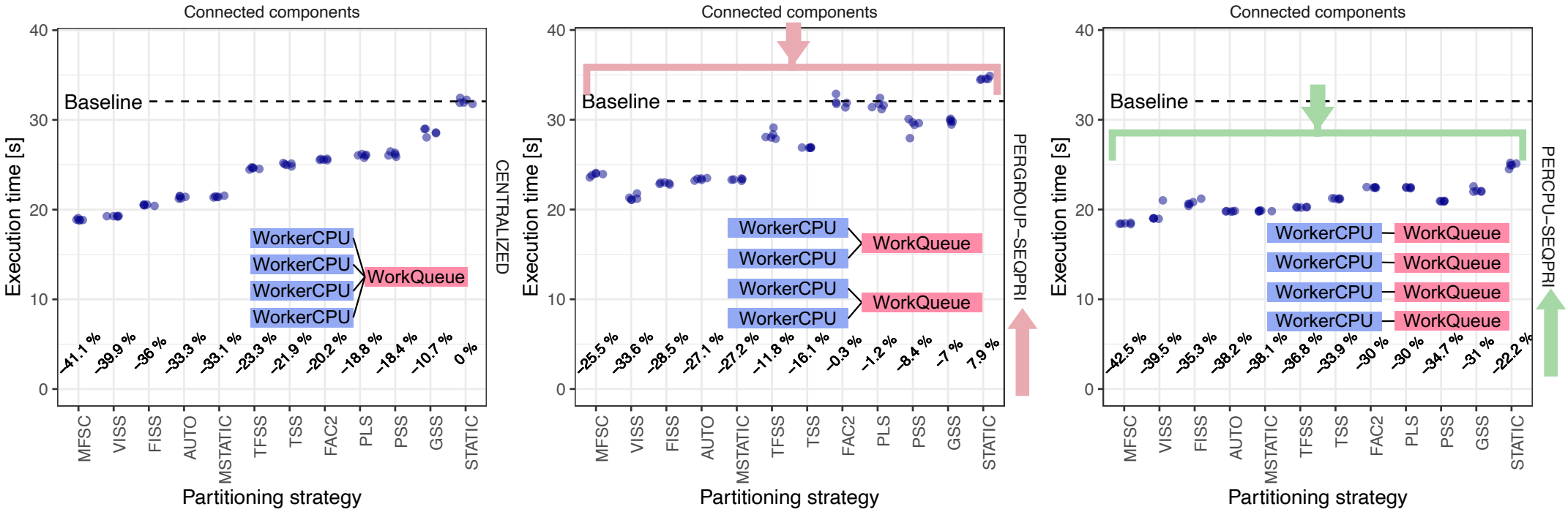
Results **CC, Broadwell, 20** threads
Local DaphneSched: Queue Layout

DAPHNE

Matrix	Size	Density
Wikipedia-20051105	$1'634'989^2$	$7.38 * 10^{-3}$

Increased parallelism management

Improved data locality
Fewer locks, less contention

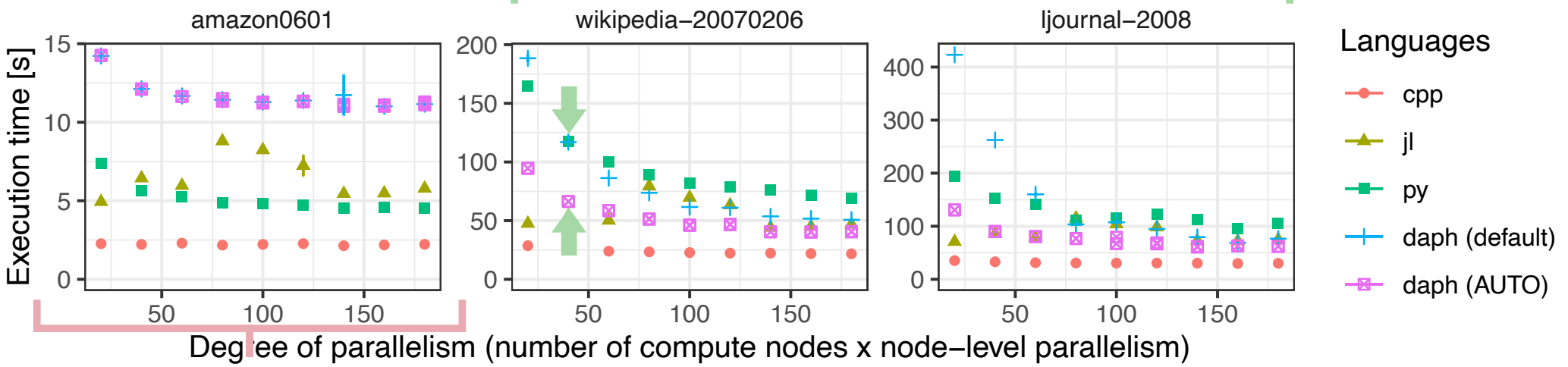


Results **CC, Broadwell**
Local & Distributed DaphneSched
VS.
C++, Python, Julia

DAPHNE

Matrix	Size	Density
amazon0601	$403'394^2$	$2.08 * 10^{-3}$
wikipedia-20070206	$3'566'907^2$	$3.54 * 10^{-4}$
ljournal-2008	$5'363'260^2$	$2.75 * 10^{-4}$

DAPHNE outperforms Python and Julia for larger input sizes with seamless effort from developers



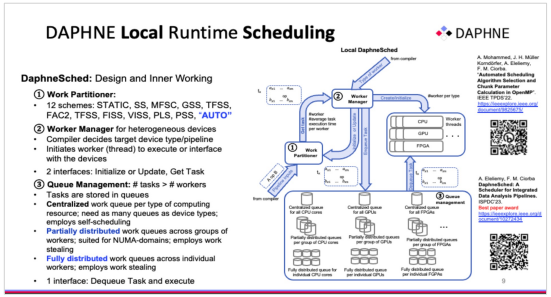
Python and Julia outperform DAPHNE for smaller inputs

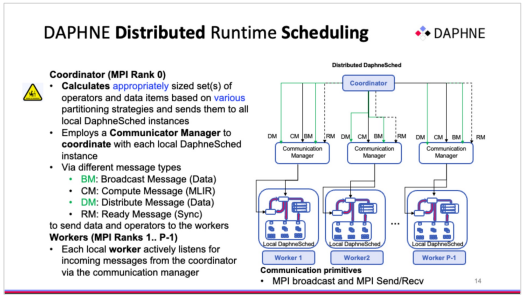
Local scheduler still relevant even on distributed executions

Takeaways

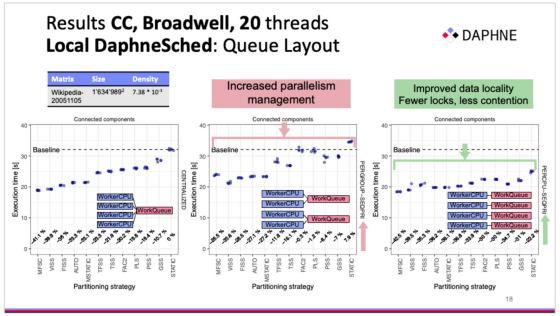
DAPHNE

DaphneSched: a **versatile, extensible, high performing, and large-scale** scheduling infrastructure for IDA pipelines (local, distributed)

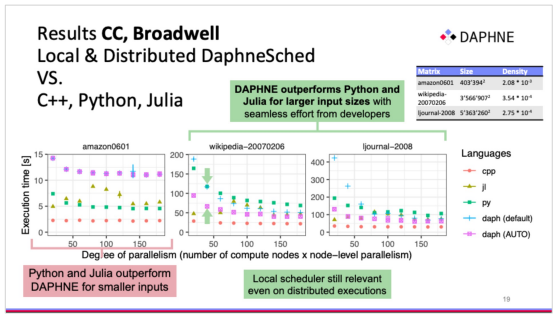




Queue layout: tug of war between management of parallelism and locality, across schemes.



DAPHNE achieves performance comparable to **other languages** with lower implementation effort



No “global best” configuration for any factor & combination. **Runtime adaptation needed.**
➔ Daphnext proposal pending.