Hewlett Packard
Enterprise

GASPI/GPI-2 – A FEW LESSONS LEARNED

Christian Simmendinger, Nils Imhoff

June 16, 2022

Global
Address Space
Programming Interface
GASPI

GASPI at a Glance

Nuts and Bolts for Communication Engines

WHY WE AIMED FOR A NEW PGAS API

Exascale assumptions back in 2010

Scaling assumptions

- 100.000 nodes, 1000 cores per node, +100 million threads.
- Strong scaling rather than weak scaling.
- Task graph models, async. dataflow computation.

User assumptions

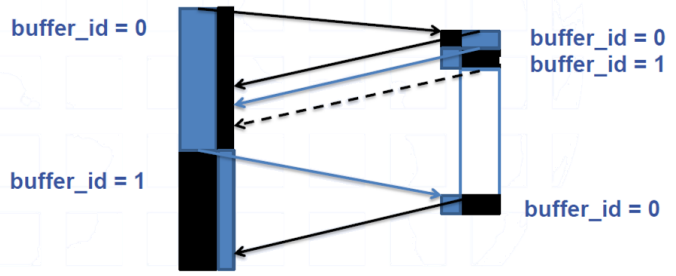
- Once we reach exascale, many HPC developers for sure will be experts.
- Framework developers + Application developers.
- Communcation patterns beyond MPI-1 (such as double buffering) are SotA.

MTBF assumptions

- MTBF 10 mins or so.
- Failure tolerance mandatory.

Hardware assumptions

- High message rates
- Heterogenous nodes



WHY WE AIMED FOR A NEW PGAS API

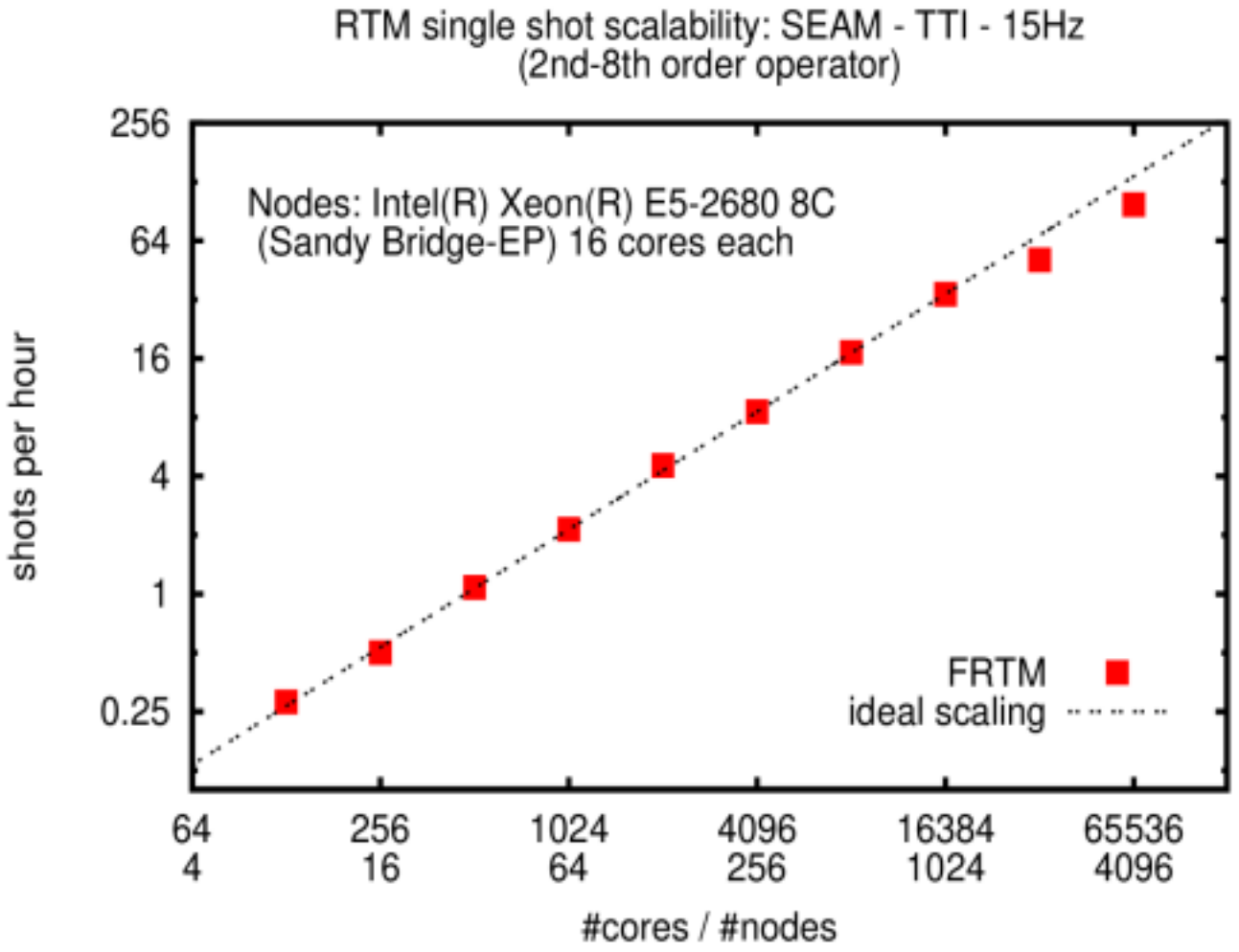
MPI did not fit.

In 2011 we started to develop GASPI/GPI-2 (German National Research Grant, Fraunhofer, T-Systems SfR, DLR, EMCL, ZIH, DWD)

- Provides a small set of core functions based on notified communication
- Enables the framework expert developers to build their own communication engines
- Provides explicit resource management, queues, requests, memory, notifications, collectives
- Aims for a memory abstraction layer, but does not assume a symmetric memory allocation.
- Supports failure tolerance.
- Supports dynamic node sets.
- Interoperates with MPI.

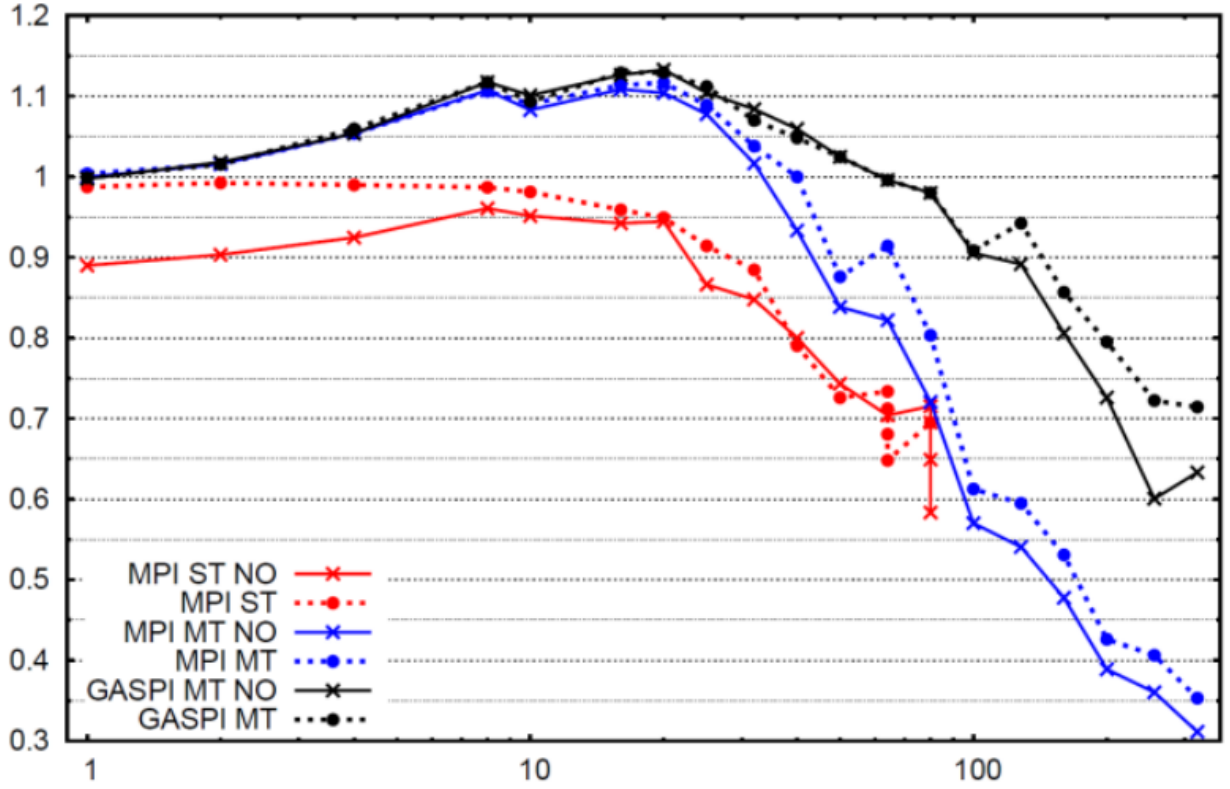
APPLICATIONS – STRONG SCALING

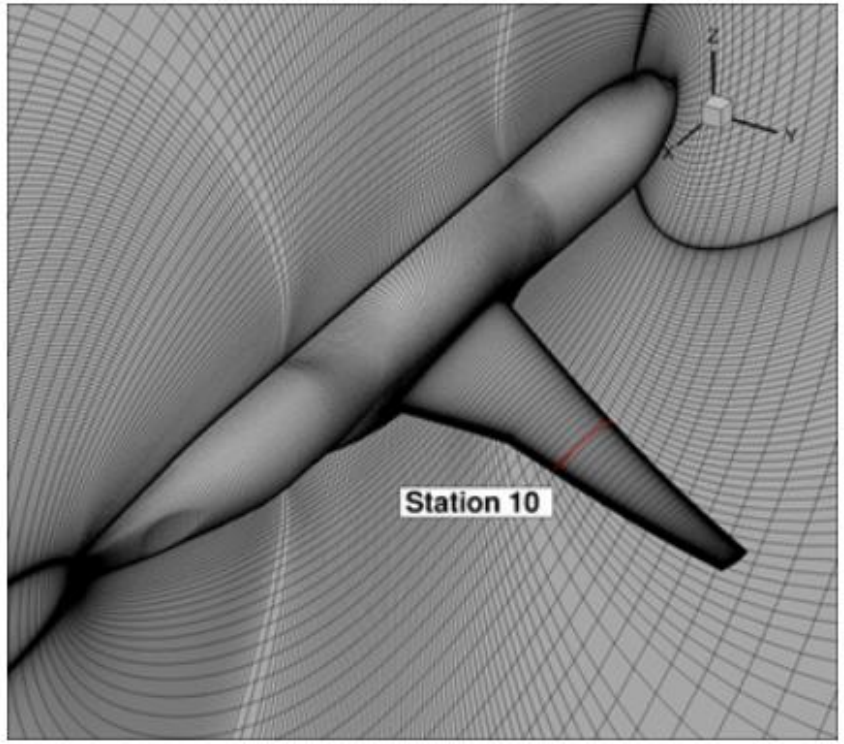
- Kirchhoff Migration - RTM (ITWM Fraunhofer) – strong scaling over 3 orders of magnitude



APPLICATIONS – STRONG SCALING

- CFD – FLUCS (DLR) - strong scaling down to 200 mesh points per core. Renamed 2018 as CODA. (DLR, ONERA, Airbus)





NEXT GENERATION CFD SOLVER 'FLUCS' T. Leicht , D. Vollmer , J. Jägersküpper , A. Schwöppe , R. Hartmann , J. Fiedler , T. Schlauch (DLR)

APPLICATIONS – PIPELINED COLLECTIVES

- Pipelined Ring implementations

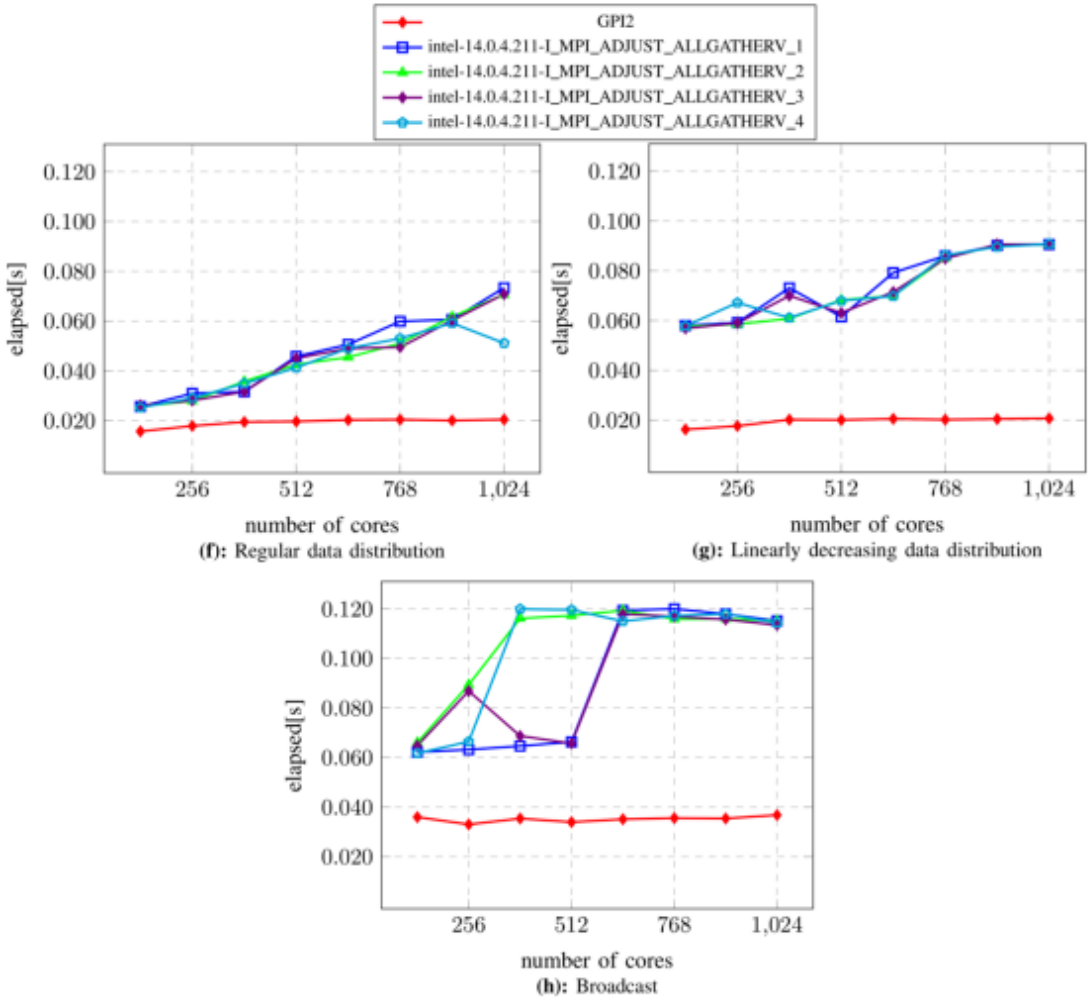
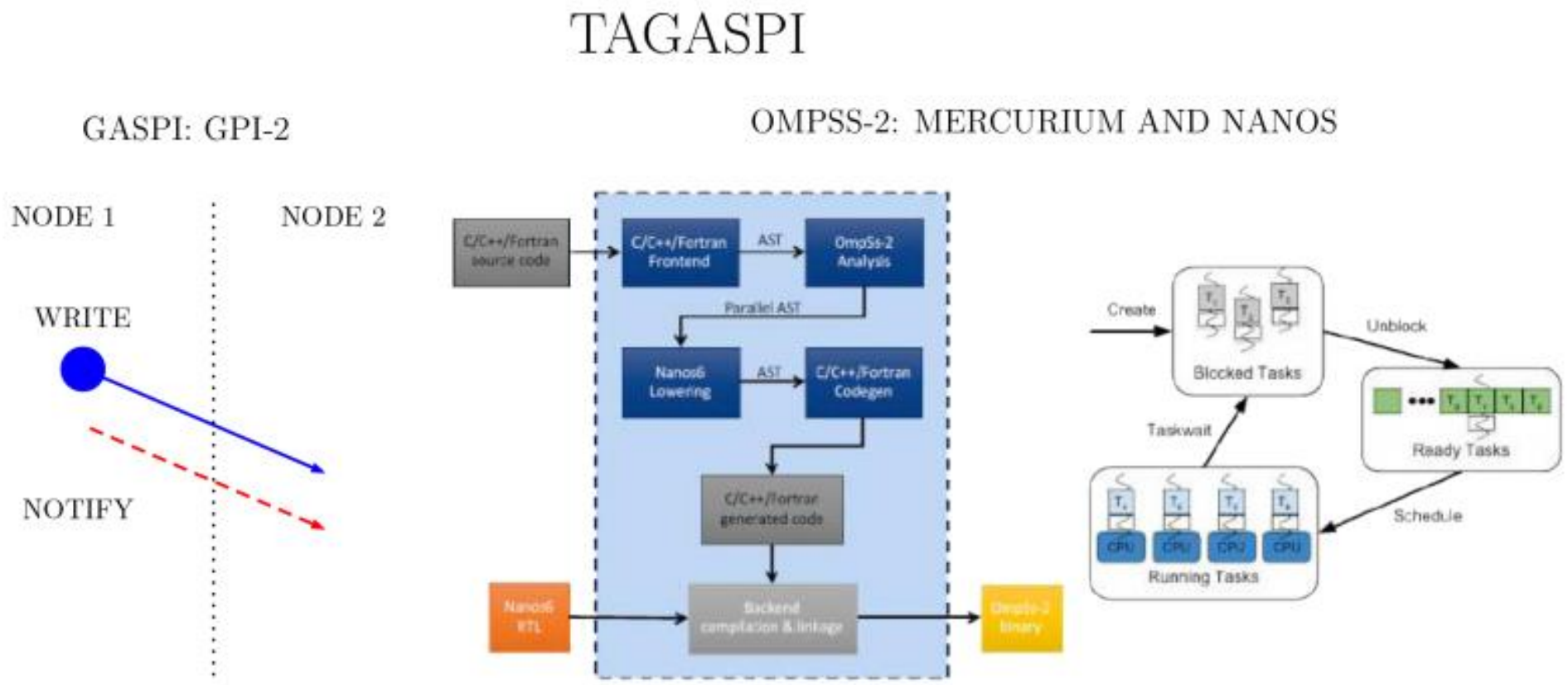


Figure 4: Performance results of Allgather(V).

APPLICATIONS – TASK GRAPH ENGINES

- OMPSS2 Intgeration - TAGASPI (BSC/ITWM) – Task Graph implementation



A FEW LESSONS LEARNED

- Provides a small set of core functions based on notified communication ☑
- Enables the framework expert developers to build their own communication engines ☑☒
- Provides explicit resource management, queues, requests, memory, notifications, collectives … ☑☒
- Aims for a memory abstraction layer, but does not assume a symmetric memory allocation. ☑
- Supports failure tolerance. ☒
- Supports dynamic node sets. ☑☒
- Interoperates with MPI. ☑

CURRENT STATUS - NEXT STEPS

- **CoE HPE/HLRS**, MPI-5.x (R.Rabenseifner, J.Gracia, C.Niethammer, J.Schuchart, R.Graham, M.Raymond, W.Okuna, N.Imhoff, C.Simmendinger) 2020-2024.
- **Efficient Notifications for MPI One-Sided Applications** (Marc Sergent, Célia Tassadit Aitkaci, Pierre Lemarinier, Guillaume Papauré, ATOS)

- Implementation for **HLRS Hawk**, **HDR200 9D EHC.** ☑
- Low level benchmarks, communication kernels ☑
  -> See talk Nils.

Applications

- Pipelined Collectives ☑
- Relativistic Quantum Field Theory (K. Szabo et. al) “Leading hadronic contribution to the muon magnetic moment from lattice QCD” (Nature 2021) ☑

THANK YOU

nils.imhoff@hpe.com, christian.simmendinger@hpe.com