TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing (ZIH)

Scalable High-Quality 1D Partitioning

2014 International Conference on High Performance Computing & Simulation (HPCS 2014)

21 – 25 July 2014, Bologna, Italy

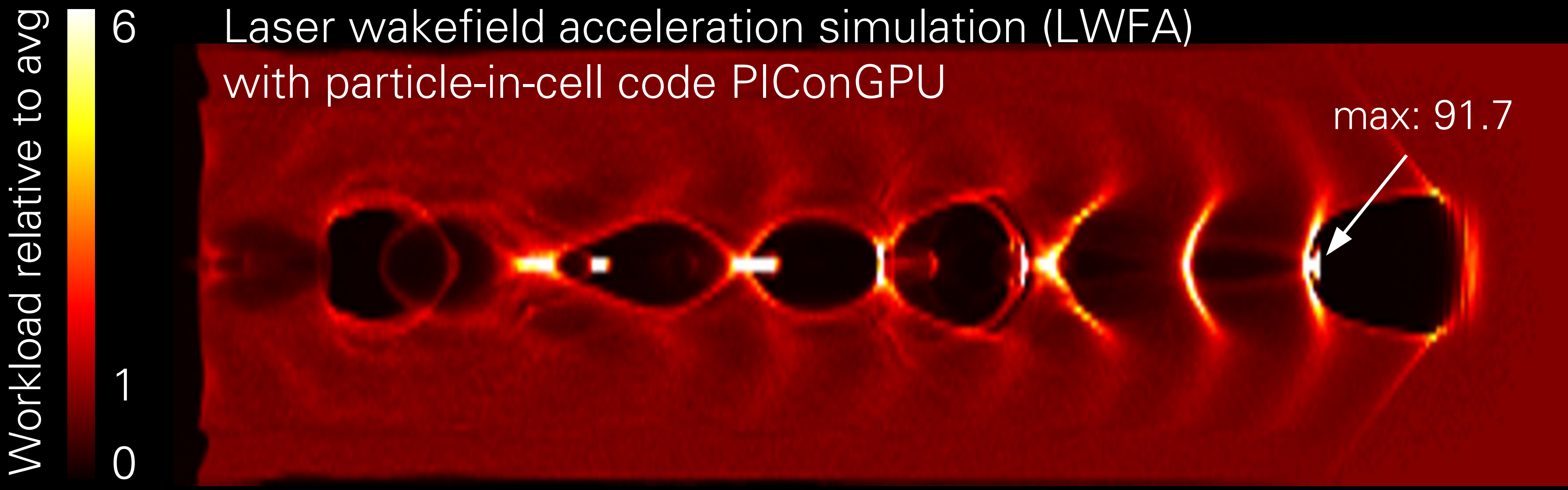
Matthias Lieber (matthias.lieber@tu-dresden.de)

Wolfgang E. Nagel

Center for Information Services and High Performance Computing (ZIH)
Technische Universität Dresden, Germany

Center for Information Services & High Performance Computing

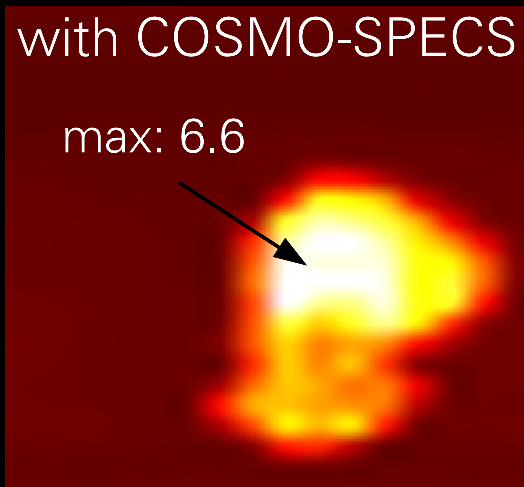
Load balance is a major challenge for scalable HPC applications, especially if the workload changes dynamically.



Laser wakefield acceleration simulation (LWFA)
with particle-in-cell code PIConGPU

Sources of workload variations:
- Adaptive spacial grids (e.g. AMR)
- Adaptive time stepping techniques
- Complex models for physical phenomena

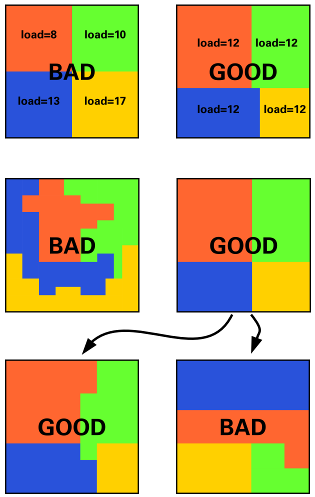
Common solution: dynamic load balancing
- Requires (frequent) repartitioning of domain



Cloud simulation
with COSMO-SPECS

Repartitioning

- Four objectives of repartitioning algorithm
  - Balance workload
  - Reduce communication between partitions (due to data dependencies)
  - Reduce migration, i.e. communication when changing the partitioning
  - Compute partitioning as fast as possible
- Contradictory goals
- Existing methods (heuristics) provide different trade-offs between the four objectives
  - Bisection methods, space-filling curves, graph methods, diffusion methods, ...

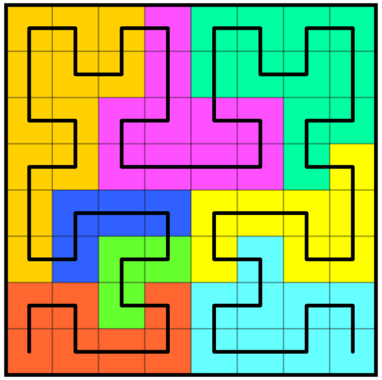


Teresco, Devine, Flaherty, *Partitioning and Dynamic Load Balancing for the Numerical Solution of Partial Differential Equations*, LNCSE, vol. 51, pp. 55-88, 2006.

From SFC Partitioning to 1D Partitioning

- Space-filling curve (SFC) partitioning widely used
  - nD space is mapped to 1D by SFC
  - Mapping is fast and has high locality
  - Migration typically between neighbor ranks
- 1D partitioning is core problem of SFC partitioning
  - Decomposes task chain into consecutive parts
- Two classes of existing 1D partitioning algorithms:
  - Heuristics: fast, parallel, no optimal solution
  - Exact methods: slow, serial, but optimal

Dynamic applications need fast, parallel, and high-quality methods to master Exascale!



Hilbert SFC

Pilkington, Baden, *Dynamic partitioning of non-uniform structured workloads with spacefilling curves*, IEEE T. Parall. Distr., vol. 7, no. 3, pp. 288-300, 1996.

Pinar, Aykanat, *Fast optimal load balancing algorithms for 1D partitioning*, J. Parallel Distr. Com., vol. 64, no. 8, pp. 974-996, 2004.

Overview

- 1D Partitioning Problem Definition
- State of the Art 1D Partitioning
- Improving an Exact Method
- Scalable High-Quality 1D Partitioning
- Conclusion

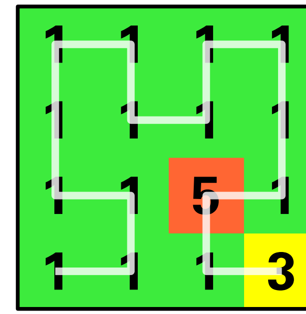
1D Partitioning Problem Definition

Decompose a vector w_i of N positive task weights into P consecutive partitions while minimizing the maximum load (i.e. bottleneck B) among the partitions.

Example for N=16 tasks and P=4 partitions:

- Task weight vector w_i ($i=1,2,\dots,N$):

1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	3



- Optimal solution:

1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	3

- Load L_p of the partitions:

$L_0 = 6$	$L_1 = 6$	$L_2=6$	$L_3=4$

  – Maximum load / Bottleneck $B = \max(L_p) = 6$

  – Load balance $\Lambda_{opt} = (\sum w_i / P) / B_{opt} = 0.92$ ($\Lambda = 1$ would be perfect)

State of the Art 1D Partitioning: Heuristics

- Recursive bisection commonly used
- Heuristic H2 by Miguet and Pierson better to parallelize
- Example for H2 with N=16, P=4, task weights w_i:

1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	3

(1) Prefix sum $W_j = \sum_{i=1}^{j} w_i$

1	2	3	4	5	6	7	8	9	10	11	12	13	18	19	22

(2) First task of partition p: min s with $W_s > p\,(W_N / P)$

$L_0 = 5$	$L_1 = 6$	$L_2 = 2$	$L_3 = 9$

(3) Increment s if W_s is closer to $p\,(W_N / P)$ than $W_{(s-1)}$

$L_0 = 5$	$L_1 = 6$	$L_2 = 7$	$L_3 = 4$

- Bottleneck $B = \max(L_p) = 7$
- Load balance $\Lambda = (W_N / P) / B = 5.5 / 7 = 0.79$

Recursive Bisection Heuristic:
Oden, Patra, Feng, *Domain Decomposition for Adaptive hp Finite Element Methods*, Contemp. Math., vol. 180, 1994.

Method described here:
Miguet, Pierson, *Heuristics for 1D rectilinear partitioning as a low cost and high quality answer to dynamic load balancing*, LNCS, vol. 1225, 1997, pp. 550-564.

State of the Art 1D Partitioning: Exact Methods

- Various exact methods have been proposed
- Fastest method by Pinar and Aykanat: ExactBS
- Based on binary search for the optimal bottleneck B_{opt}
- Search Interval: $W_N / P \leq B_{opt} \leq B_{Heuristic}$
- Requires probing whether a partition exists for given B
  - Binary search on W_j for each P: O(P log N)
  - Han et al.: O(P log (N/P))
  - Pinar and Aykanat: O(P log P + P log(w_{max}/w_{avg}))
- Problems
  - Slower than heuristics
  - More fatal: serial only, one process needs to collect task weights from all other processes

Extensive Overview: Pinar, Aykanat, *Fast optimal load balancing algorithms for 1D partitioning*, J. Parallel Distr. Com., vol. 64, no. 8, pp. 974-996, 2004.

Han, Narahari, Choi, *Mapping a chain task to chained processors*, Inform. Process. Lett., vol. 44, no. 3, pp. 141-148, 1992.

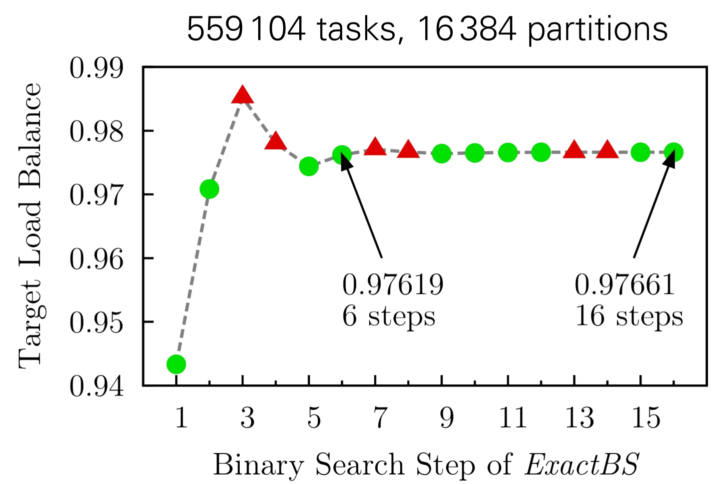
Improving an Exact Method: ExactBS → QBS → QBS*

(1) New probe algorithm, tuned for small N / P

- Guess that consecutive partitions have same number of tasks, if not, do linear (!) search on W_j starting from guessed position

(2) QBS: Quality-Assuring Bisection Algorithm

- Stop binary search at given quality q
- Guarantees load balance $\Lambda = q\ \Lambda_{opt}$
- Reduces number of search steps
- Exact method for q = 1

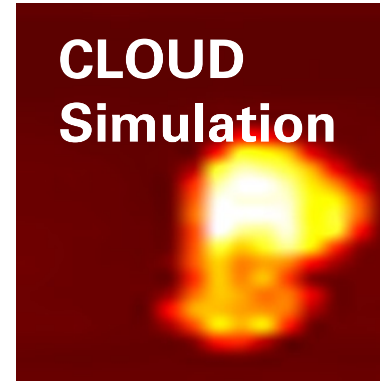


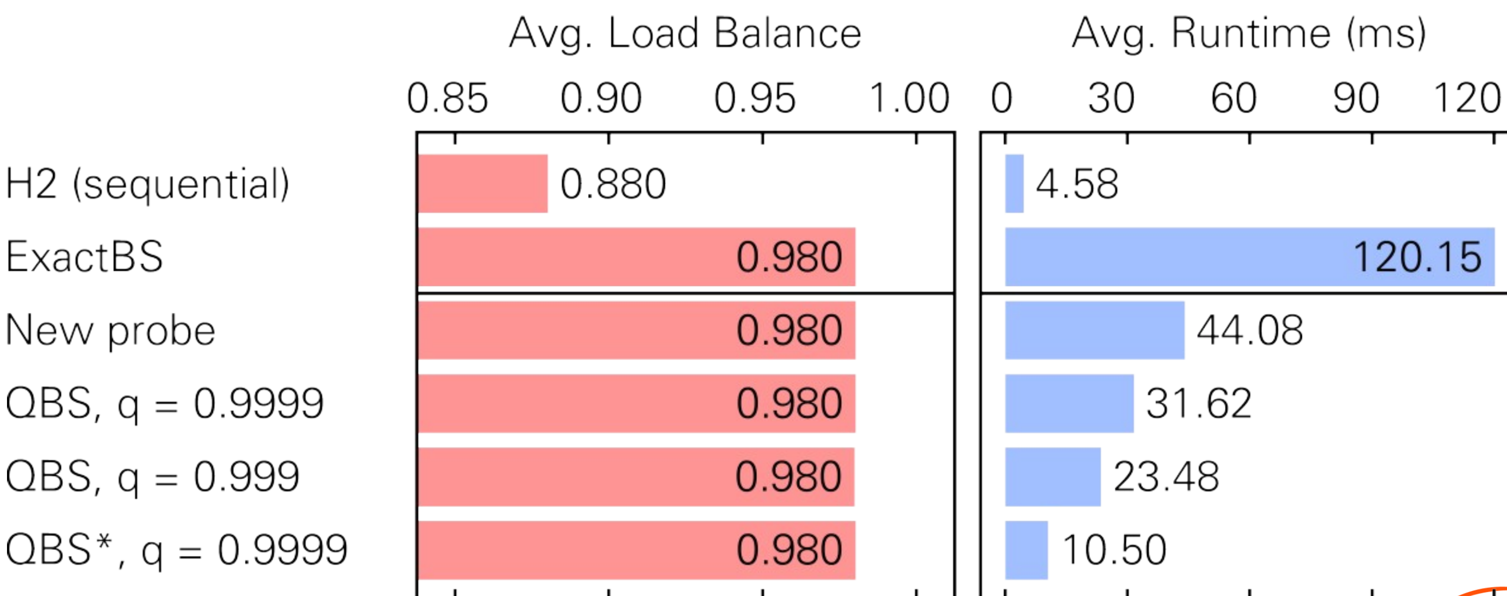
(3) QBS*: Parallelization of binary search

- Search interval for Bopt is split between processes
- Reduces number of search steps
- Downside: all processes need task weights of all processes

Improving an Exact Method: Comparison Benchmark

- Comparison to existing methods H2 and ExactBS
- Averages over 100 successive task weight vectors from the cloud simulation
- System: JUQUEEN, IBM Blue Gene/Q
- 559 104 tasks, 16 384 partitions / MPI ranks





Runtime includes only partitioning calculation, i.e. no collection of task weights and no prefix sum

Scalable High-Quality 1D Partitioning: Algorithm HIER*

Large scale applications require a fully parallel method, i.e. without gathering all task weights

- Run parallel H2 to create G < P coarse partitions:

Orig part. | P0 | P1 | P2 | P3 | P4 | P5 | P6 | P7

↯Parallel Heuristic H2

Coarse part. | P0 / P1 | P2 / P3 | P4 / P5 | P6 / P7

- Run G independent instances of exact QBS* (q=1.0) to create final partitions within each group:

↯QBS* | ↯QBS* | ↯QBS* | ↯QBS*

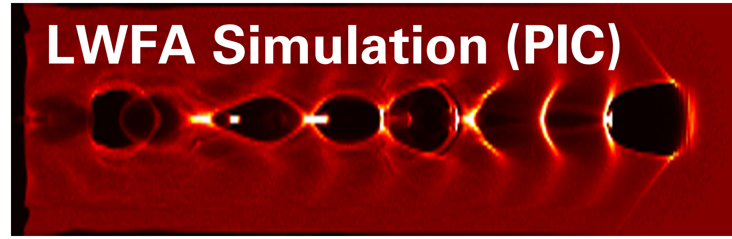
Final part. | P0 | P1 | P2 | P3 | P4 | P5 | P6 | P7

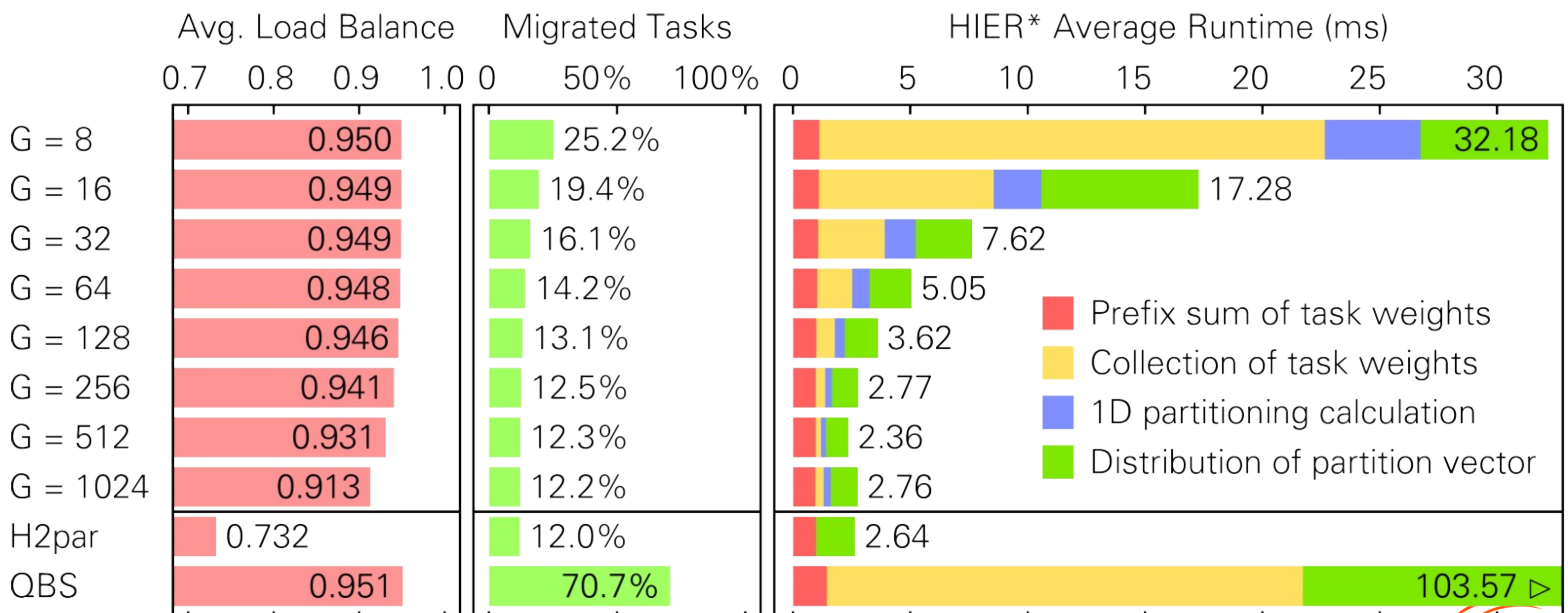
- Parameter G allows trade-off between scalability (high G → heuristic dominates) and load balance (small G → exact method dominates)

H2 nearly optimal if $w_{max} << W_N / P$: Miguet, Pierson, *Heuristics for 1D rectilinear partitioning as a low cost and high quality answer to dynamic load balancing*, LNCS, vol. 1225, 1997, pp. 550-564.

Scalable High-Quality 1D Partitioning: Group Count G

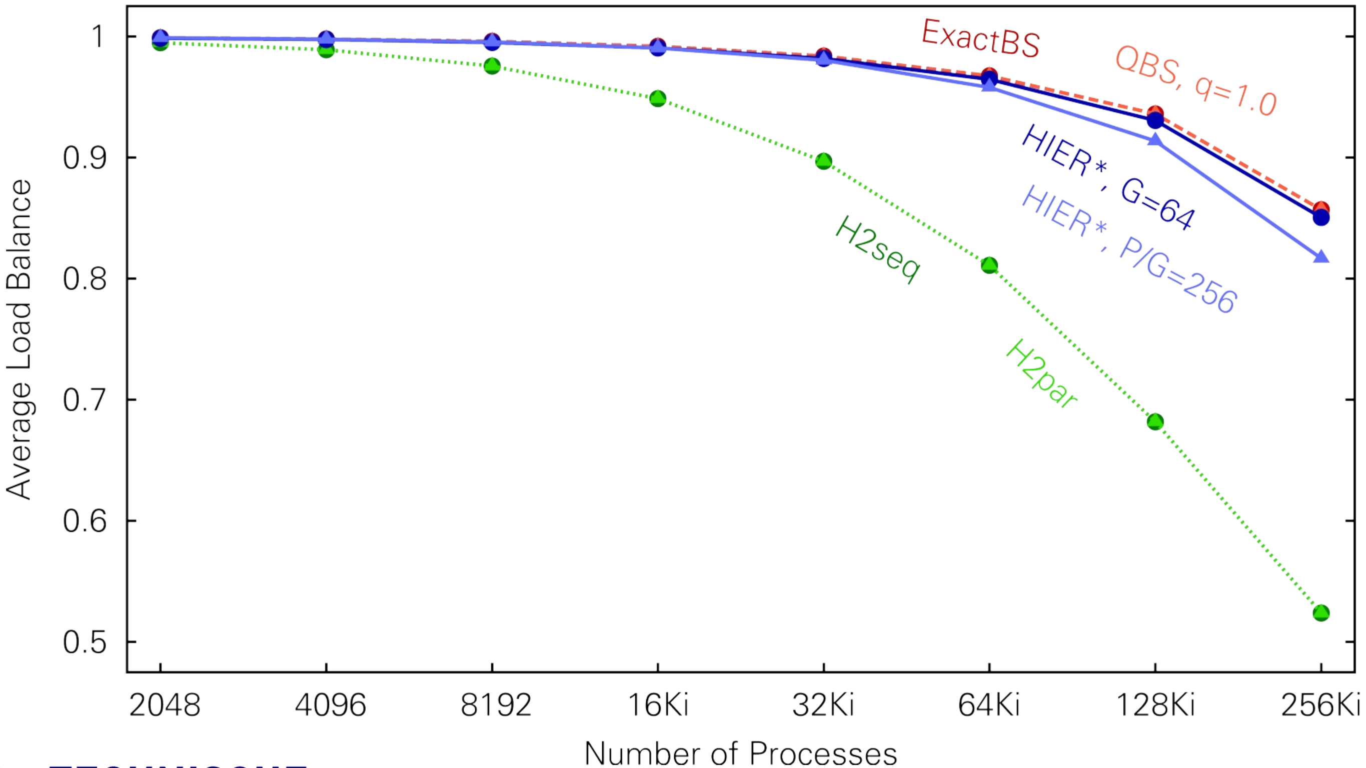
- Comparison of HIER* with parallel heuristic H2 and QBS
- Averages over 2000 successive task weight vectors of the LWFA simulation
- System: JUQUEEN, IBM Blue Gene/Q
- 1 048 576 tasks, 16 384 partitions / MPI ranks





Scalable High-Quality 1D Partitioning: Load Balance

- Cloud simulation, 1 357 824 tasks
- System: JUQUEEN, IBM Blue Gene/Q
- HIER*, G=64 achieves 99.2 % of the optimal load balance at 262 144 processes

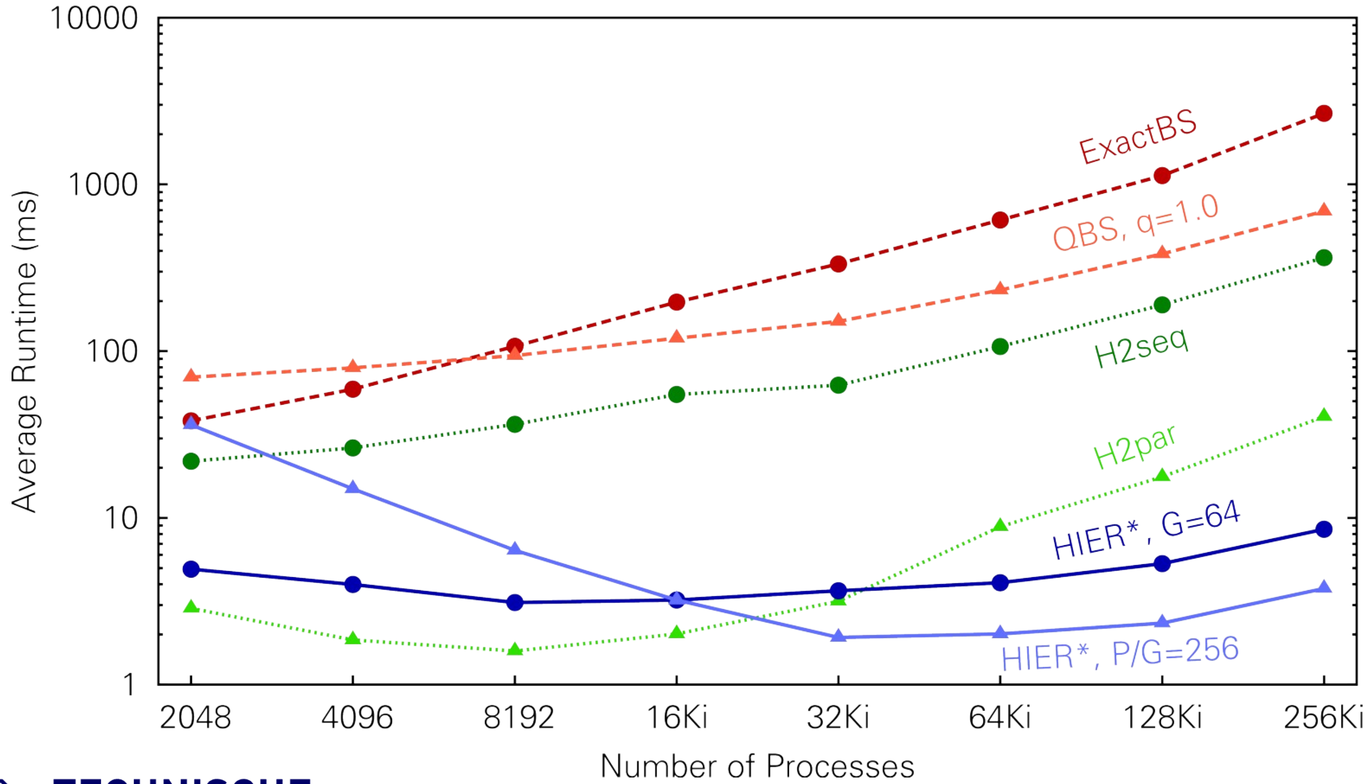


TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services & High Performance Computing

Scalable High-Quality 1D Partitioning: BG/Q Scalability

- Cloud simulation, 1 357 824 tasks
- System: JUQUEEN, IBM Blue Gene/Q
- HIER*, G=64 runs at 262 144 processes ~300x faster than ExactBS



ExactBS: 2668 ms

QBS: 692 ms

H2seq: 363 ms

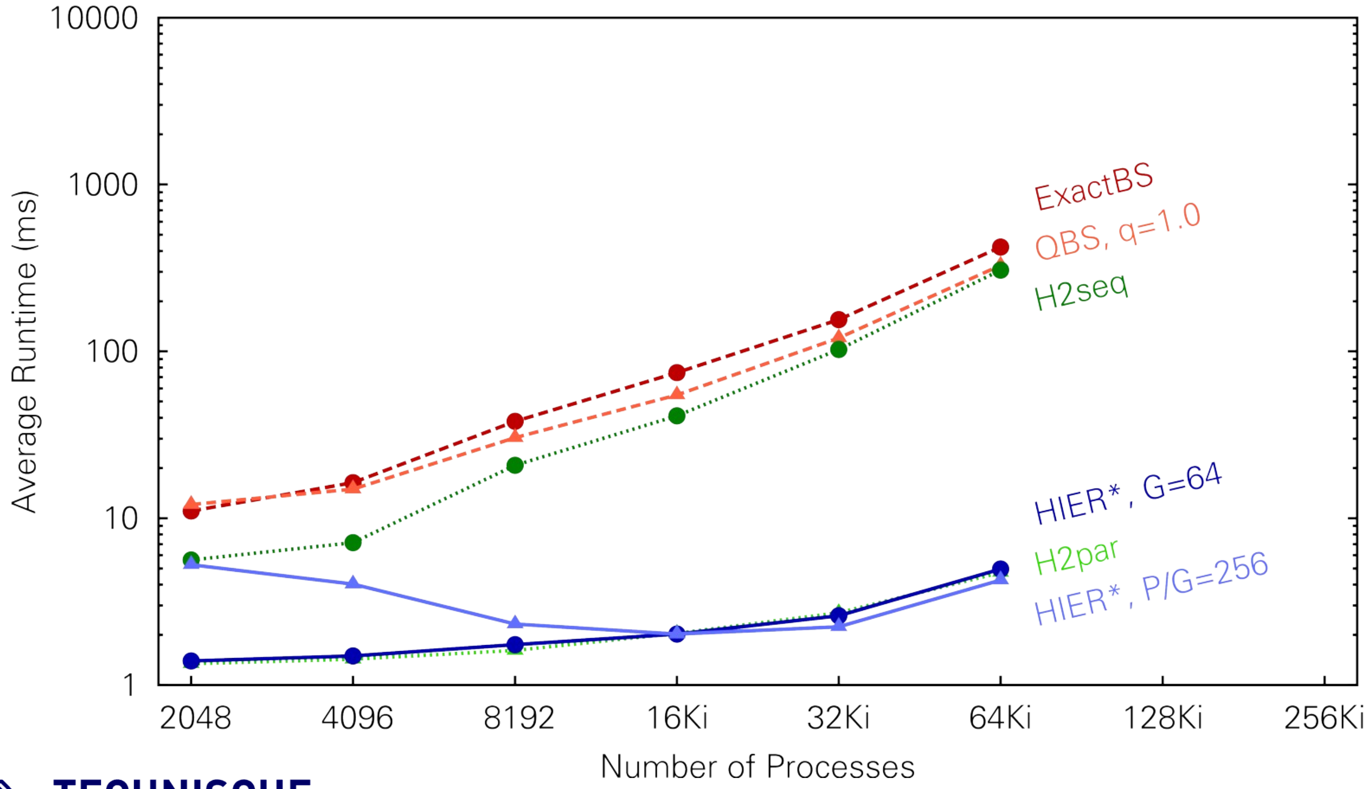
H2par: 40.5 ms

$\text{HIER*}_{G=64}$: 8.55 ms

$\text{HIER*}_{P/G=256}$: 3.77 ms

Scalable High-Quality 1D Partitioning: iDataPlex Scalability

- Cloud simulation, 1 357 824 tasks
- System: SuperMUC, IBM iDataPlex (E5-2680, IB FDR10)
- HIER*, G=64 runs at 65 536 processes ~85x faster than ExactBS



ExactBS: 421 ms

QBS: 328 ms

H2seq: 307 ms

$\text{HIER}^*_{G=64}$: 4.96 ms

H2par: 4.73 ms

$\text{HIER}^*_{P/G=256}$: 4.26 ms

TECHNISCHE UNIVERSITÄT DRESDEN

ZIH Center for Information Services & High Performance Computing

Conclusions

- Optimized published exact 1D partitioning algorithm
- Developed scalable, hierarchical algorithm
- Implemented in the open source dynamic load balancing and model coupling framework FD4
  - Benchmark program, and cloud dataset available to reproduce results
  - Enables dynamic load balancing up to 262 144 processes for COSMO-SPECS+FD4
- Future work
  - Comparison using other applications workload data
  - 1D partitioning algorithms tuned for low migration
  - Avoid replication of full partition vector on all ranks

FD4 website and benchmark download: http://wwwpub.zih.tu-dresden.de/~mlieber/fd4

Lieber, Nagel, Mix, *Scalability Tuning of the Load Balancing and Coupling Framework FD4*, NIC Symposium 2014, pp. 363-370.

Thank you very much for your attention!

Acknowledgments

Verena Grützun, Ralf Wolke,
Oswald Knoth, René Widera,
Daniel Hackenberg

lrz Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities

VAMPIR www.vampir.eu

TROPOS
Leibniz Institute for Tropospheric Research

DWD COSMO CONSORTIUM FOR SMALL SCALE MODELING

PIConGPU

HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

www.tropos.de

www.cosmo-model.org

picongpu.hzdr.de

Funding

DFG Deutsche Forschungsgemeinschaft

cfaed CENTER FOR ADVANCING ELECTRONICS DRESDEN

Europa fördert Sachsen.
ESF Europäischer Sozialfonds

IMDATA

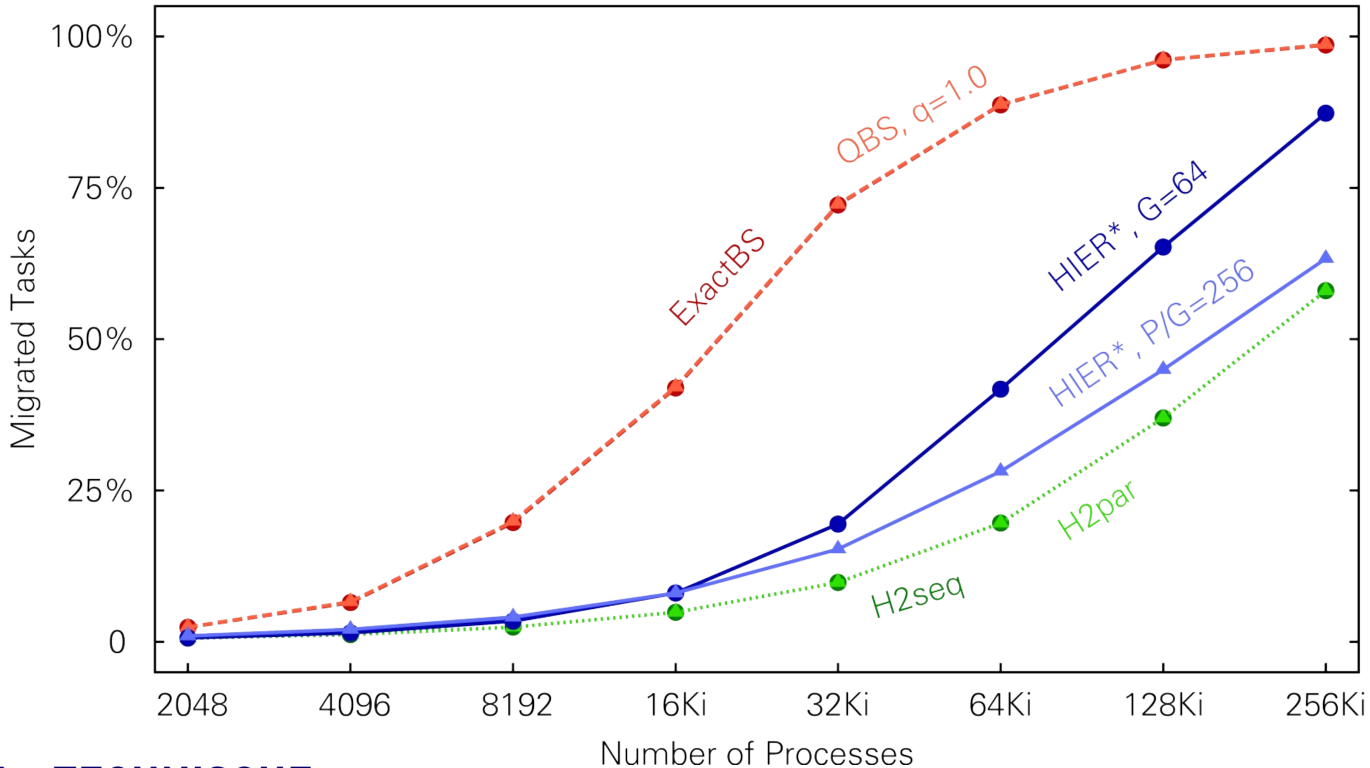
TECHNISCHE UNIVERSITÄT DRESDEN

ZIH Center for Information Services & High Performance Computing

Backup Slides

Scalable High-Quality 1D Partitioning: Migration

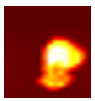
- Cloud simulation, 1 357 824 tasks
- System: JUQUEEN, IBM Blue Gene/Q
- Hierarchical method lies between exact and heuristic



TECHNISCHE UNIVERSITÄT DRESDEN

ZIH Center for Information Services & High Performance Computing

QBS and QBS* Performance (16 384 processes)



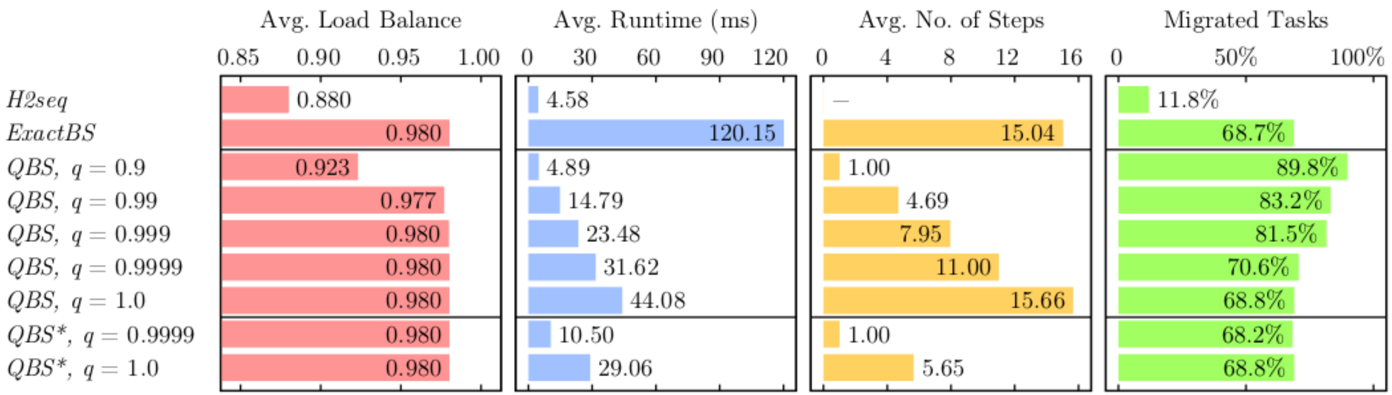


Fig. 5. 1D partitioning benchmark results for the sequential algorithms *H2seq*, *ExactBS*, and *QBS* with the CLOUD dataset (559 104 tasks) for 16 384 processes on JUQUEEN. *QBS** is a version with parallelized search for the optimal bottleneck. The runtime includes the 1D partitioning calculation only.

HIER* Algorithm as seen from MPI

- Prefix sum of task weights + broadcast of total load
  - *MPI_Exscan* (parallel prefix sum)
  - MPI_Bsend + MPI_Recv (consistency at partition borders)
  - *MPI_Bcast* (last rank broadcasts total load to all)
- Compute coarse partitioning with parallel H2
  - MPI_Isend + MPI_Recv + MPI_Waitall (send partition borders to group master)
  - MPI_Bcast (group master broadcasts borders within group)
- Collection of task weights within groups
  - MPI_Isend + MPI_Irevc + MPI_Waitall (send weights from other groups ranks to the nearest ranks in the group these tasks belong to in coarse partitioning)
  - MPI_Allgather (exchange task weights within group)
- Exact partitioning within group with QBS*, q=1
  - MPI_Allreduce
- Distribition of global partition vector
  - MPI_Allgather (between all group masters)
  - MPI_Bcast (within groups)

HIER and HIER* Performance (16 384 processes)

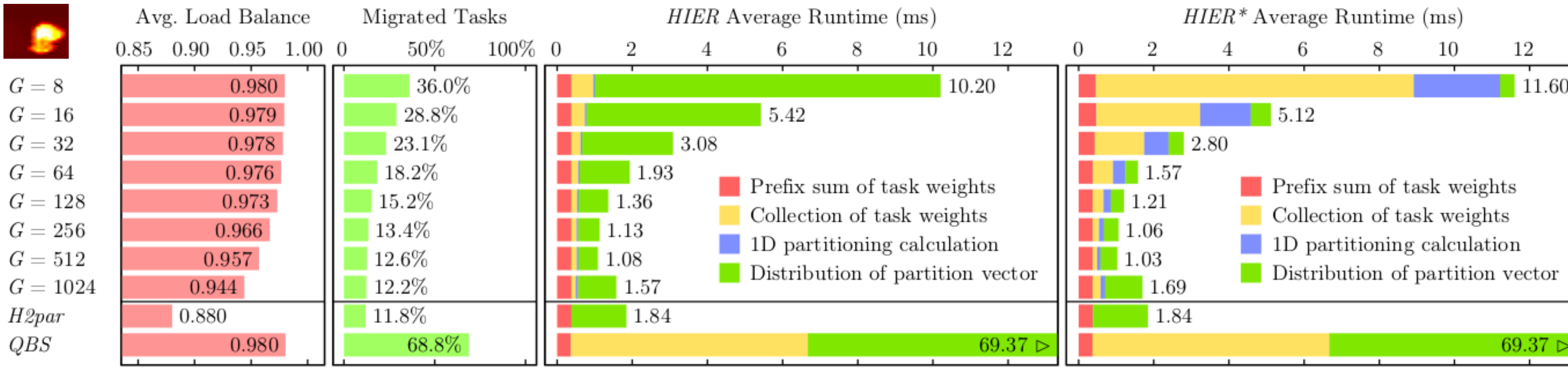


Fig. 6. 1D partitioning benchmark results of the hierarchical methods *HIER* and *HIER** with the CLOUD dataset (559 104 tasks) for 16 384 processes on JUQUEEN. For comparison, the results of the parallel heuristic *H2par* and the sequential exact algorithm *QBS* ($q = 1$) are shown.

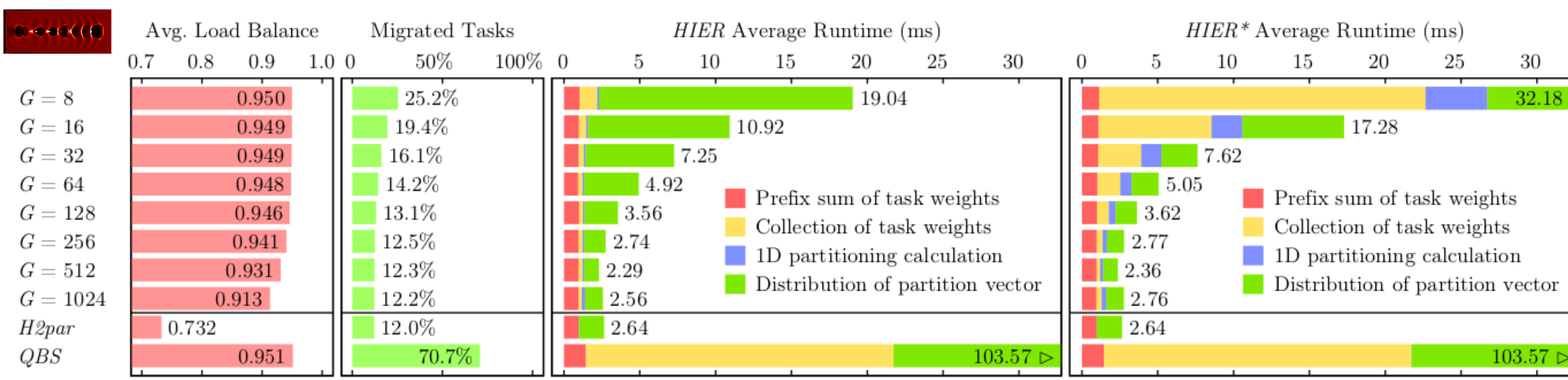
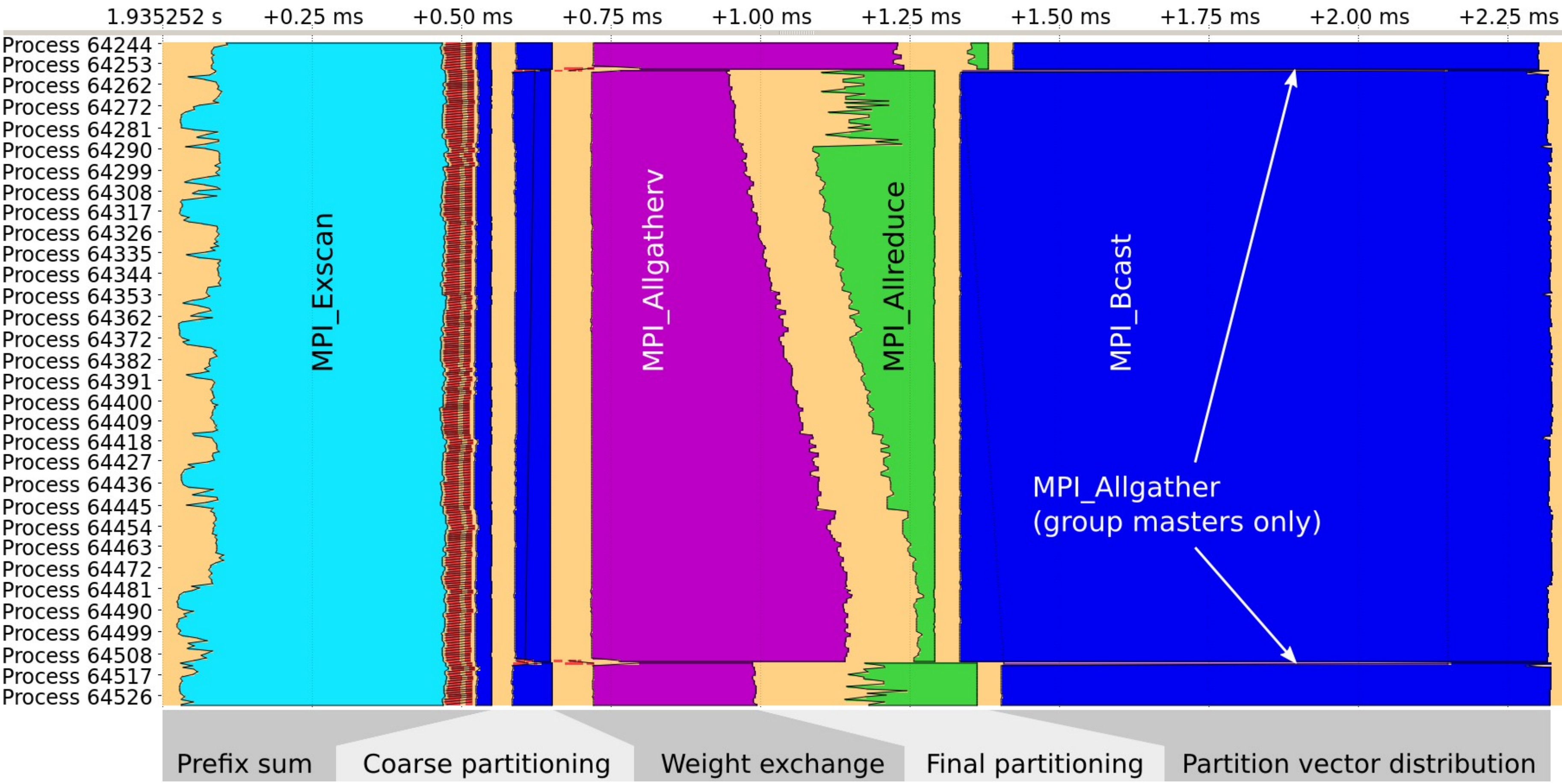


Fig. 7. 1D partitioning benchmark results of the hierarchical methods *HIER* and *HIER** with the LWFA dataset (1 048 576 tasks) for 16 384 processes on JUQUEEN. For comparison, the results of the parallel heuristic *H2par* and the sequential exact algorithm *QBS* ($q = 1$) are shown.

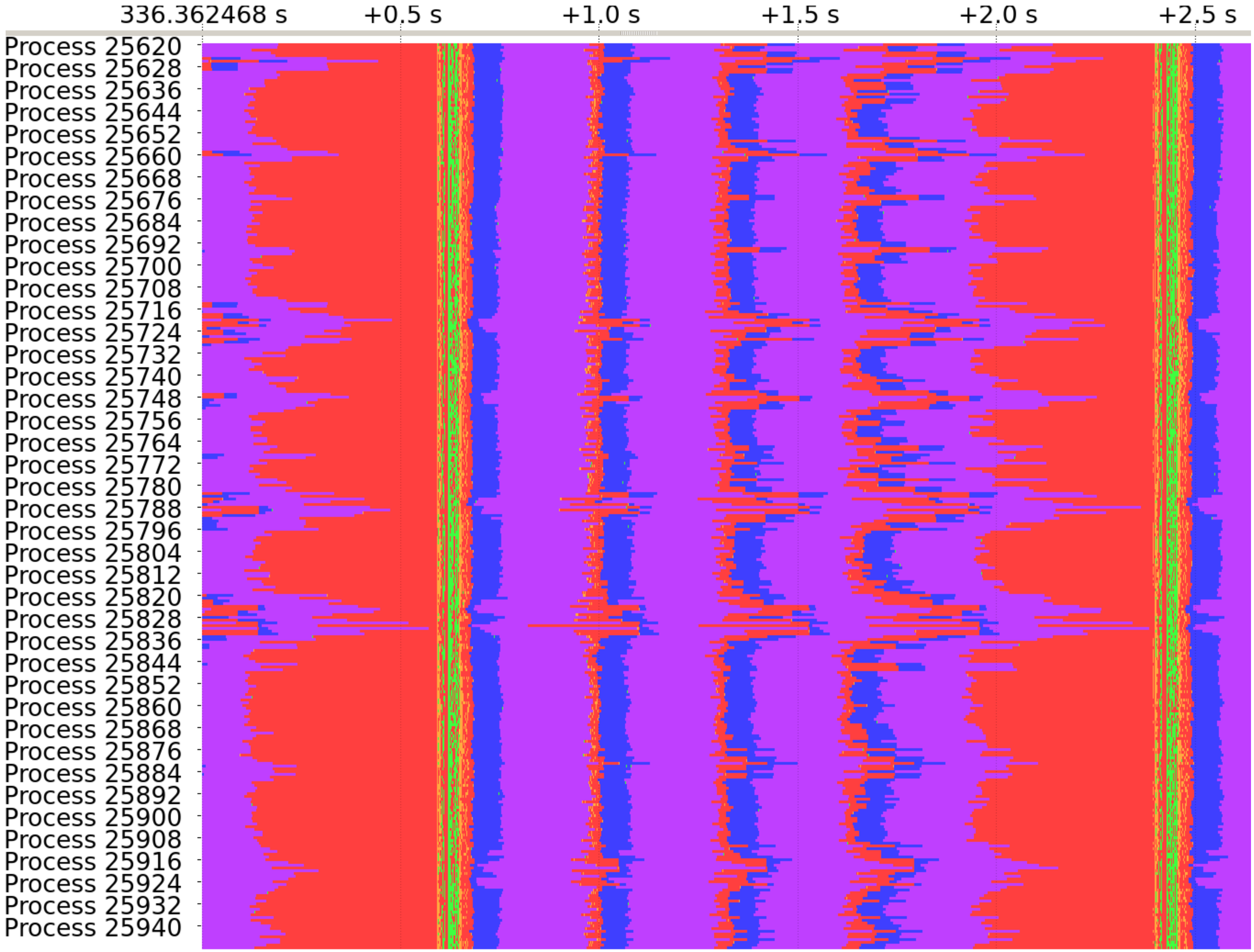
HIER* seen in Vampir (one Group of 256 out of 64Ki)



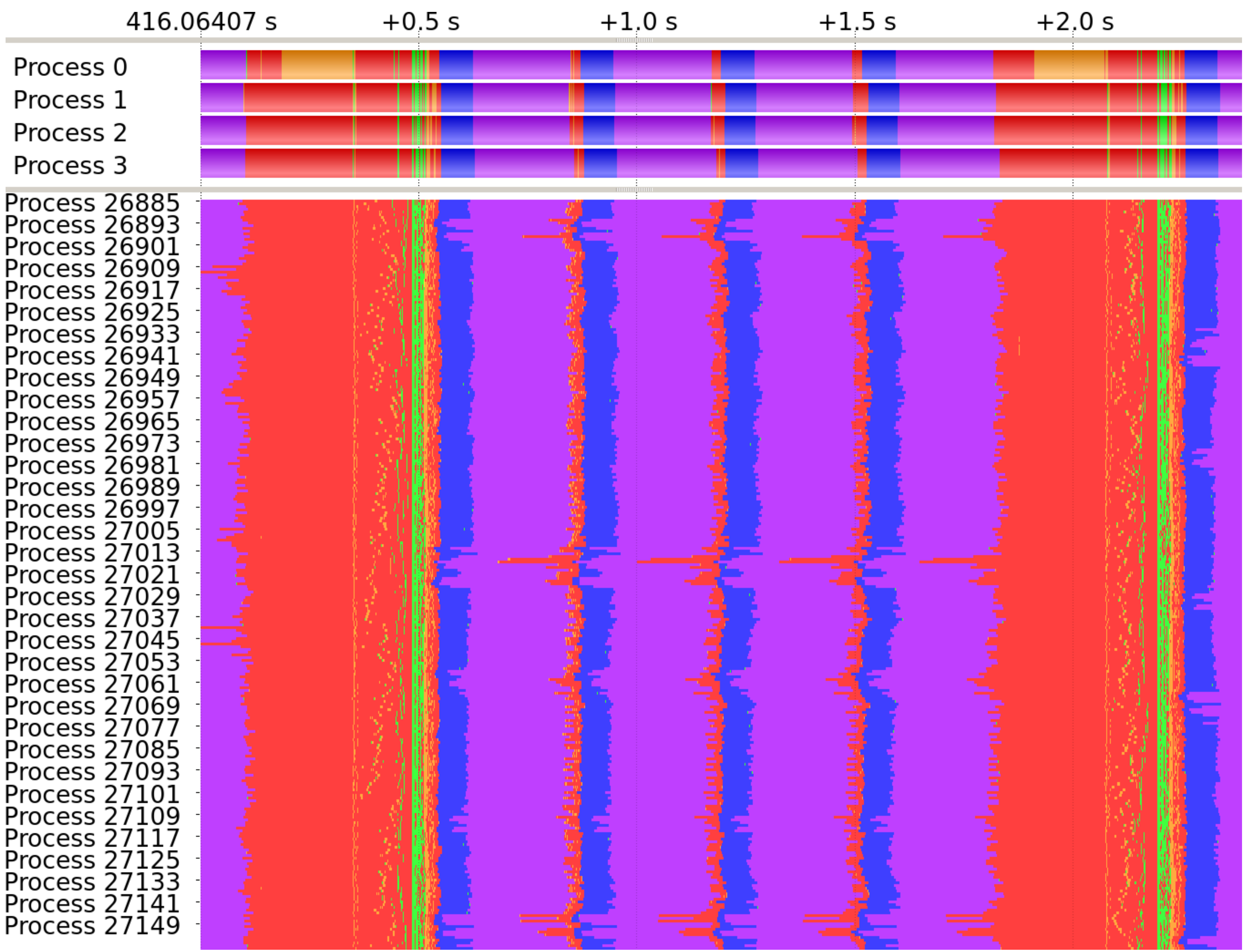
TECHNISCHE UNIVERSITÄT DRESDEN

ZIH Center for Information Services & High Performance Computing

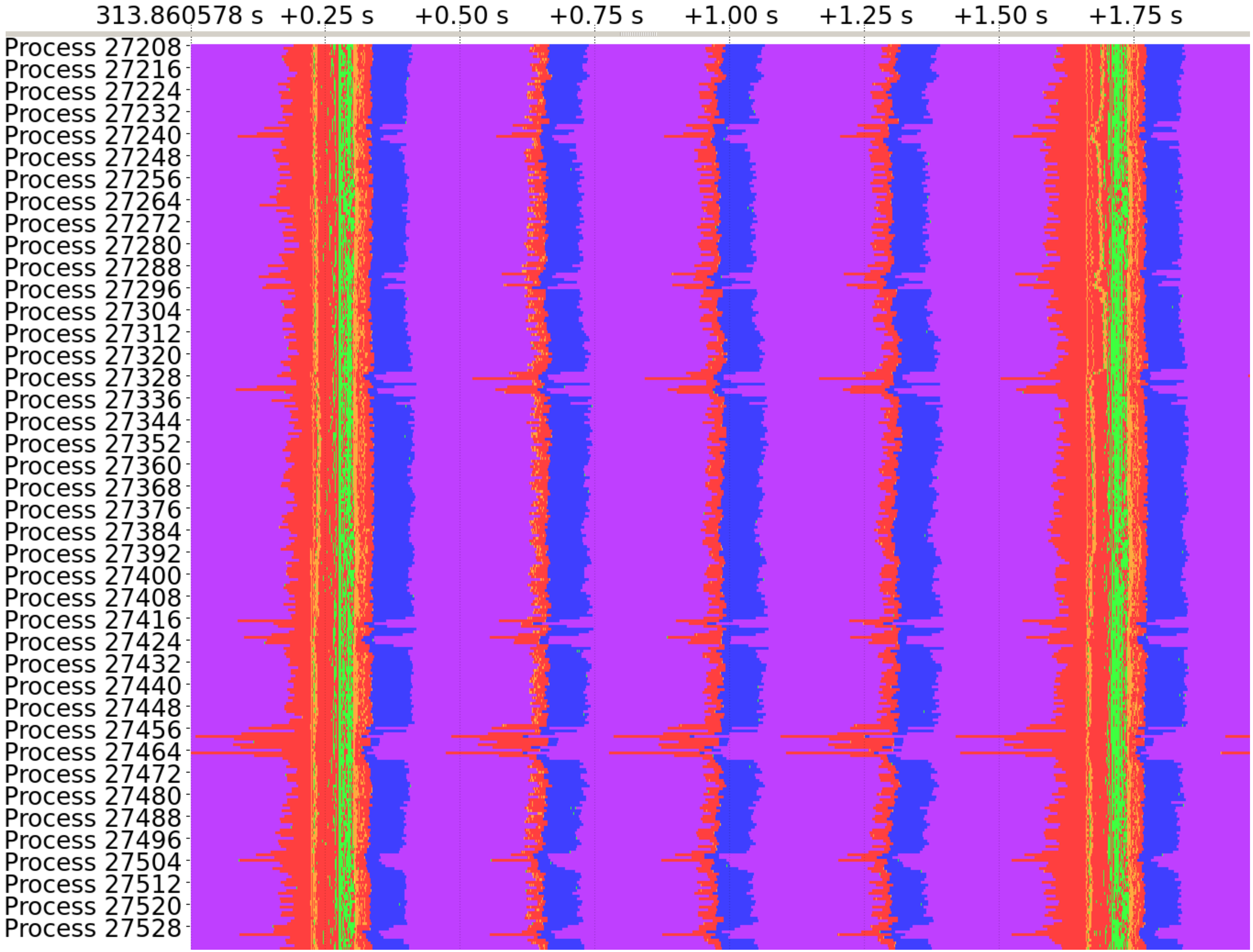
Heuristic H2 in Action (COSMO-SPECS+FD4)



ExactBS in Action (COSMO-SPECS+FD4)



HIER* in Action (COSMO-SPECS+FD4)



COSMO-SPECS+FD4: Comparison of Methods

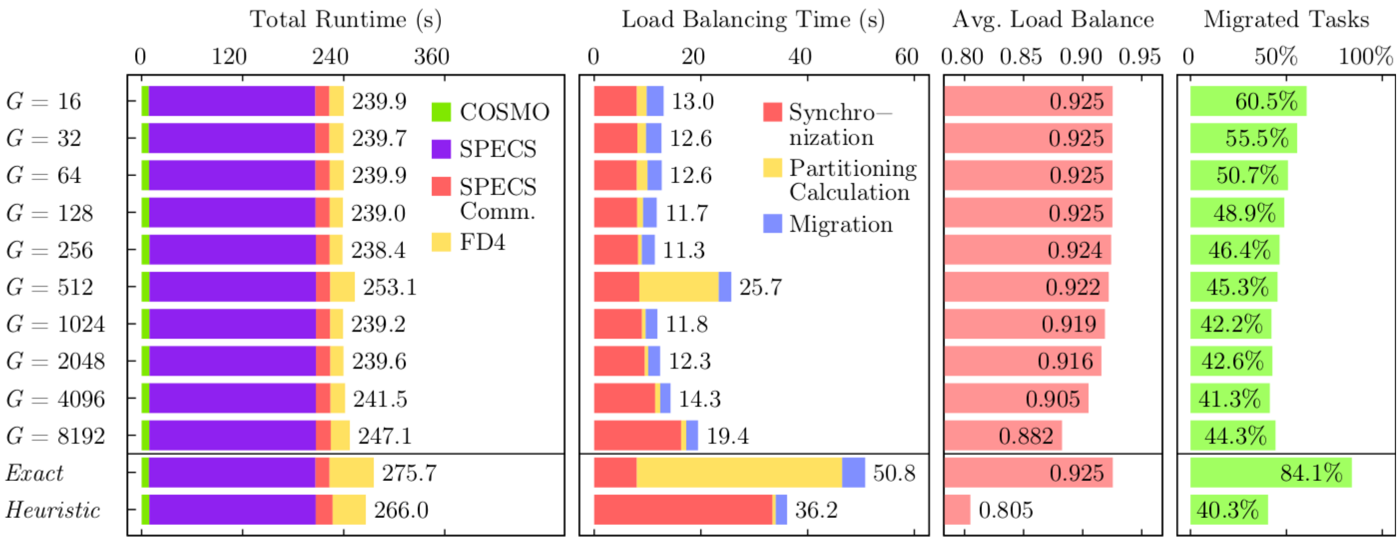


Figure 3. Influence of the group count G on the hierarchical 1D partitioning algorithm in COSMO-SPECS+FD4 with 65 536 processes on BlueGene/Q. The exact method and the heuristic are included as reference.

Lieber, Nagel, Mix, *Scalability Tuning of the Load Balancing and Coupling Framework FD4*, NIC Symposium 2014, pp. 363-370.