Bundesministerium
für Forschung, Technologie
und Raumfahrt

Less Data, Same Science?

Mitigating Data Movement Bottlenecks in Fusion Plasma Simulations with Lossy Compression

Diego Jiménez Vargas
DaREXA-F Project

MPCDF
MAX PLANCK
COMPUTING & DATA FACILITY

MAX-PLANCK-INSTITUT
FÜR PLASMAPHYSIK

FAU
Friedrich-Alexander-Universität
Erlangen-Nürnberg

ParTec
MODULAR SUPERCOMPUTING

TUM

Agenda

- **Communication as scaling bottleneck in fusion plasma simulations**
- **Lossy compression for scientific data**
  - Agnostic lossy compression landscape
- **Evaluating state-of-the-art lossy compressors**
  - Performance modeling and the "*Compressor Overhead Limit*"
- **Compressing Boundary Exchanges**
- **Application Results**
  - BSL6D
  - GENE
- **Conclusions and future work**

Data Movement as Scaling Bottleneck: GENE

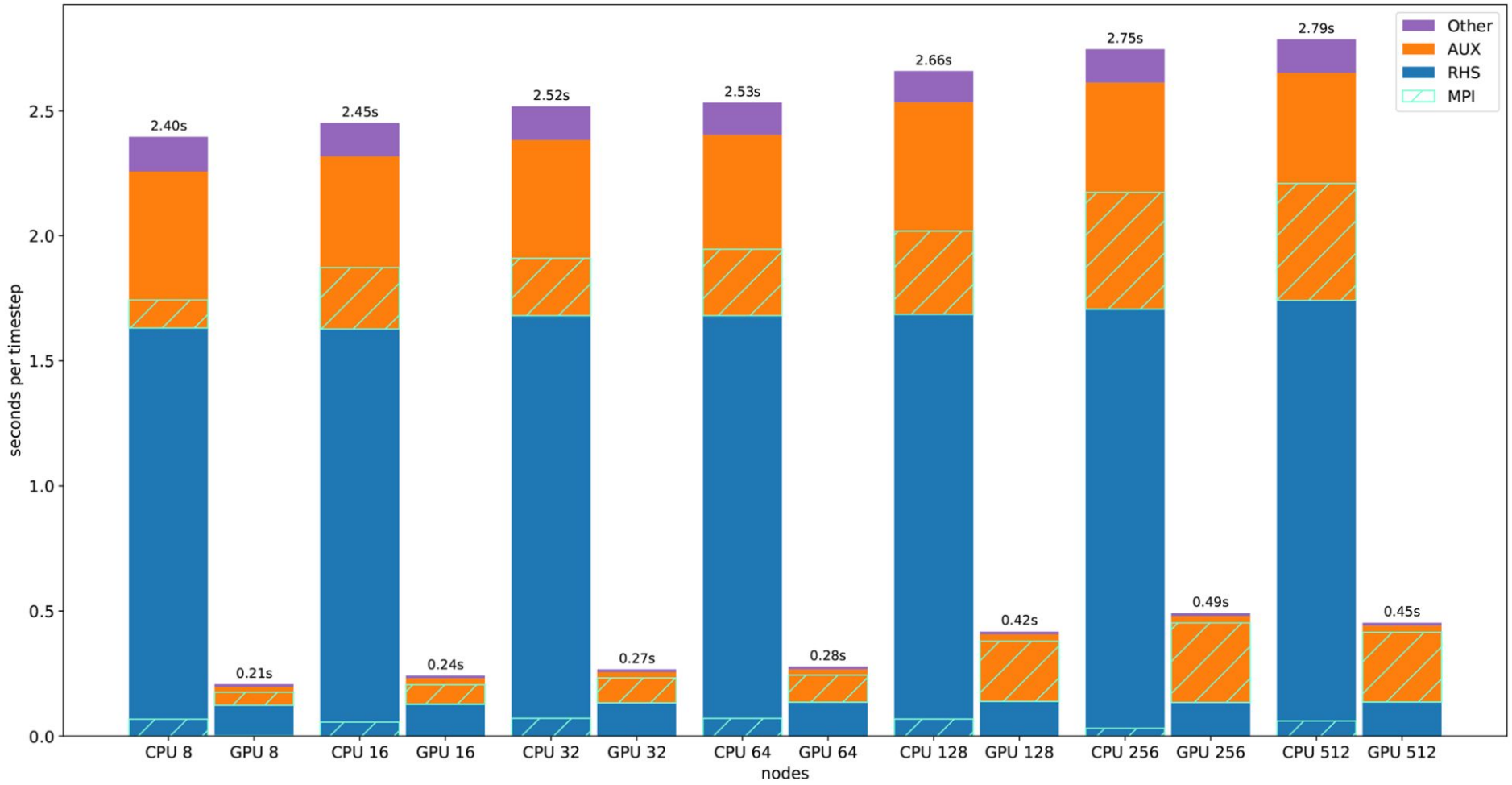


Figure: Weak scaling analysis on Summit. CPU vs GPU execution. Per node: 2 x IBM Power9 and 6 NVIDIA V100
Source: (Germaschewski, et al., Physics of Plasma, 2021)

Data Movement as Scaling Bottleneck: BSL6D

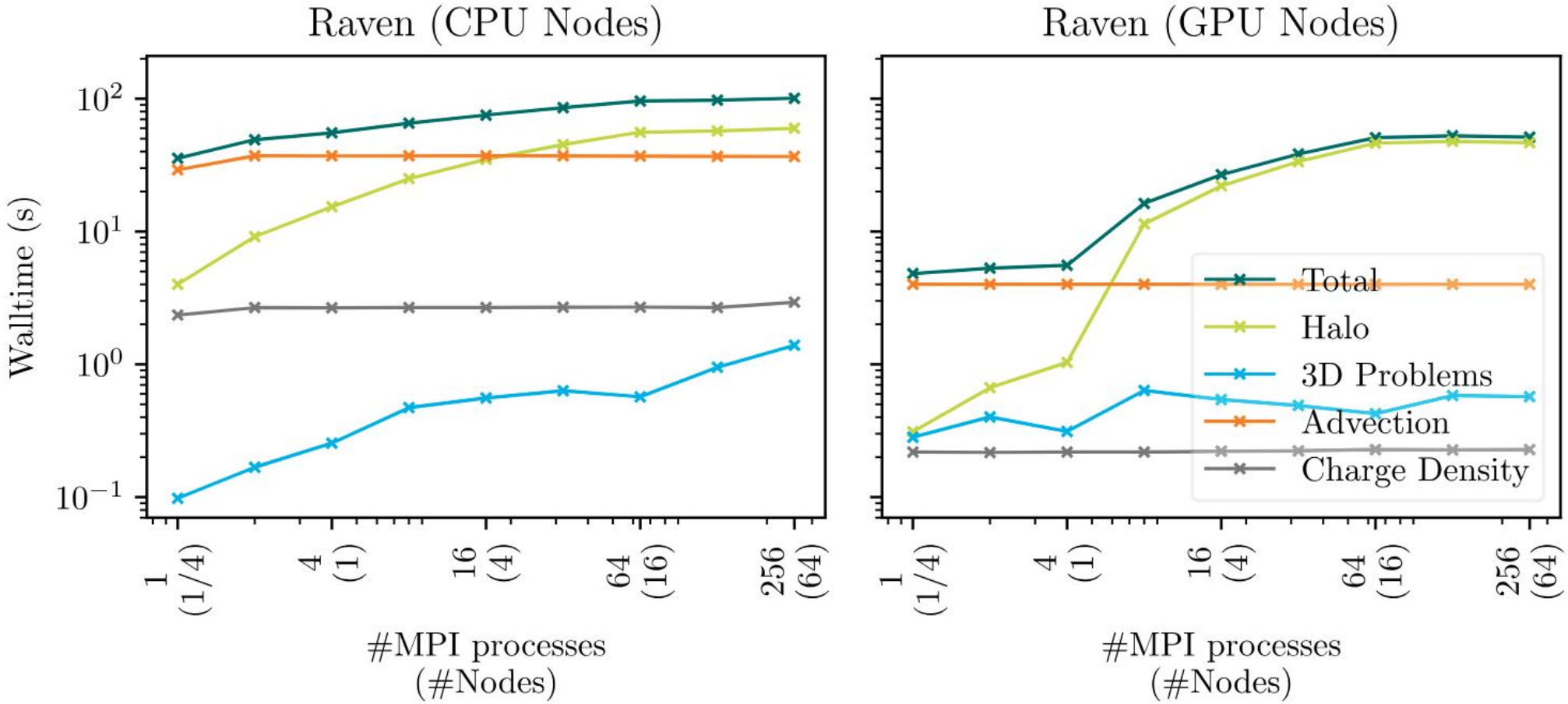


Figure: Weak scaling analysis on Raven. CPU and GPU execution. Per node: Intel Xeon IceLake Platinum 8360Y and 4 NVIDIA A100

Source: (Schild, et al., *Computer Physics Communications*, 2024)

Lossy Compression for Scientific Data

Lossy Compression for Scientific Data

Lossy Compressors	Year	Model	Compress. Pipeline	Device	Type	QoI	Performance	Compression Ratio
FPZIP [129]	2006	Pred.	PDP+BPC	CPU	S	N	*** [57]	** [57, 123]
ISABELA [108, 109]	2013	Deci.	Sorting+DS+Bspline	CPU	S	N	* [59]	* [59]
ZFP [53, 128]	2014	Trans.	DT+OWT+BPC	CPU/GPU	S	N	**** [123, 181]	****[123, 181]
NUMARCK [46]	2014	Pred.	PDP+QT	CPU	S	N	-	** [57]
SSEM [166]	2015	Trans.	DWT+QT+FTR+LE	CPU	S	N	-	** [57]
SZ0.1 [57]	2016	Pred.	PDP+LE	CPU	S	N	*** [57]	*** [57]
Bitgrooming [209]	2016	BitM.	BPC+LE	CPU	S	N	-	* [189]
SZ1.4 [181]	2017	Pred.	PDP+QT+LE	CPU	S	N	*** [123, 181]	*** [123, 181]
SZ2 [123]	2018	Pred.	PDP+QT+LE	CPU	S	N	**** [123, 217]	**** [123, 217]
MGARD [15, 18]	2018	Pred.	PDP+QT+LE	CPU	U and S	Y	** [126]	*** [126]
SpaioTmpDeci. [120]	2018	Decim.	DS	CPU	S	N	**** [120]	* [120]
Digitrounding [2]	2019	BitM.	BPC+LE	CPU	S	N	-	* [189]
DCTZ [211]	2019	Trans.	OWT+QT+LE	CPU	S	N	*** [211]	*** [211]
GhostSZ [204]	2019	Pred.	PDP+QT+LE	FPGA	S	N	*** [204]	**** [204]
TTHRESH [31]	2019	HOSVD	SVD+BPC+LE	CPU	S	N	* [24, 31, 134]	***** [24]
ZFP-V [176]	2019	Trans.	DT+OWT+BPC+LE	FPGA	S	N	**** [176]	**** [176]
SZauto [219]	2020	Pred.	PDP+QT+LE	CPU	S	N	*** [219]	**** [219]
waveSZ [185]	2020	Pred.	QT+PDP+LE	FPGA	S	N	*** [185]	**** [185]
TuckerMPI [30]	2020	HOSVD	SVD	CPU	S	N	** [24]	** [24]
zMesh[147]	2021	Pred.	DT	CPU	U	N	*** [147]	**** [147]
TEZip [165]	2021	DeepL.	DNN+LE	CPU	S	N	**** [165]	**** [165]
cuSZ [184, 186]	2021	Pred.	QT+PDP+LE	GPU	S	N	** [86]	*** [85, 86, 135]
MGARD+ [126]	2021	Pred.	PDP+QT+LE	CPU	S	N	*** [126]	**** [126]
SZ3 [118, 217]	2021	Pred.	PDP+QT+LE	CPU	S	N	*** [217]	**** [133, 134, 217]
AESZ [131, 138]	2021	DeepL.	DNN+PDP+QT+LE	CPU	S	N	** [131]	**** [131]
CAE [138]	2021	DeepL.	DNN+PDP+QT+LE	CPU	S	N	** [138]	**** [138]
MGARD-GPU[44, 68]	2021	Pred.	PDP+QT+LE	GPU	S	N	*** [68]	*** [68]
DE-ZFP[71]	2022	Trans.	DT+OWT+BPC+LE	FPGA	S	N	**** [71]	**** [71]
QoZ [132]	2022	Pred.	PDP+QT+LE	CPU	S	N	*** [132, 134]	**** [132, 134]
SZx [207]	2022	BitM.	FTR+BPC	CPU	S	N	***** [207]	** [207]
cuSZx [207]	2022	BitM.	FTR+BPC	GPU	S	N	*** [86]	** [86]
QoI-SZ3 [95]	2022	Pred.	DT+PDP+QT+LE	CPU	S	Y	** [95]	*** [95]
MGARD-Lambda [10]	2022	Pred.	PDP+QT+LE	CPU	S	Y	-	-
ATC [24]	2023	HOSVD	SVD+QT+BPC+LE	CPU	S	N	** [24]	***** [24]
ZFP-X [143]	2023	Trans.	DT+OWT+BPC+LE	CPU	S	N	**** [143]	**** [143]
SPERR [115]	2023	Trans.	DWT+QT+LE	CPU	S	N	** [115, 133]	***** [115, 133]
HH-NN [84]	2023	DeepL.	DS+DNN+QT	GPU	S	N	* [84]	***** [84]
FAZ [133]	2023	Pred.	DWT+PDP+QT+LE	CPU	S	N	** [133]	***** [133]
FZ-GPU [210]	2023	Pred.	QT+PDP+LE	GPU	S	N	**** [210]	** [210]
cuSZp [85, 86]	2023	Pred.	QT+PDP+LE	GPU	S	N	***** [85, 86]	** [85, 86]
AMR-Comp [195–199]	2023	Pred.	PDP+QT+LE	CPU	U	N	*** [197]	**** [197]
SRNN-SZ [130]	2023	Pred.	DNN+PDP+QT+LE	CPU	S	N	* [130]	***** [130]
SZ_ADT [144]	2023	Pred.	PDP+QT+LE	CPU	S	N	*** [144]	**** [144]
HPEZ(QoZ 2) [134]	2024	Pred.	PDP+QT+LE	CPU	S	N	*** [134]	**** [134]
HAE [110]	2024	DeepL.	DT+DNN+QT+FTR	GPU	S	N	-	**** [110]
SZp [13, 82]	2024	Pred.	QT+PDP+LE	GPU	S	N	***** [82]	** [82]
cuSZ-I [137]	2024	Pred.	PDP+QT+LE	GPU	S	N	**** [137]	***[137]

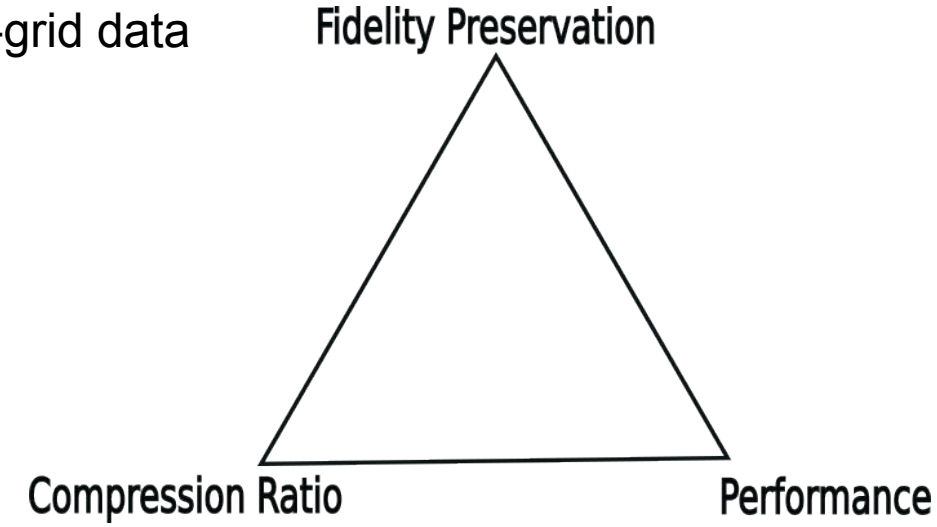
Table: Survey of general purpose lossy compressors
Source: (Di, et al., *ACM computing surveys*, 2025)

- Is certain compressor better suited for a given application?
- What compression ratios can we expect?
  - No defined theoretical bound of lossy compressibility for scientific data yet
- How will compression affect the Quantities of Interest (QoI) from the simulation?

Agnostic Lossy Compressors for Scientific Data

- **Several modern agnostic lossy compressors for floating-point data with user-defined error control**
  - **SZ [4]**: prediction-based compression
  - **ZFP [5]**: transform-based compression
  - **MGARD [6]**: multi-grid data reduction

Compressor	GPU Backends	Supported Data Types	Data Dimensionality	Striding Support	Error-Bound Control
ZFP (1.0.1)	CUDA	FP64, FP32	1D-3D	✓	×
cuSZp (V3.0.0)	CUDA	FP64, FP32	1D-3D	×	✓
MGARD-X (1.6.0)	CUDA HIP SYCL	FP64, FP32	1D-5D	×	✓



Is compressor throughput enough for inline usage?

Evaluating State-of-the-Art Lossy Compressors

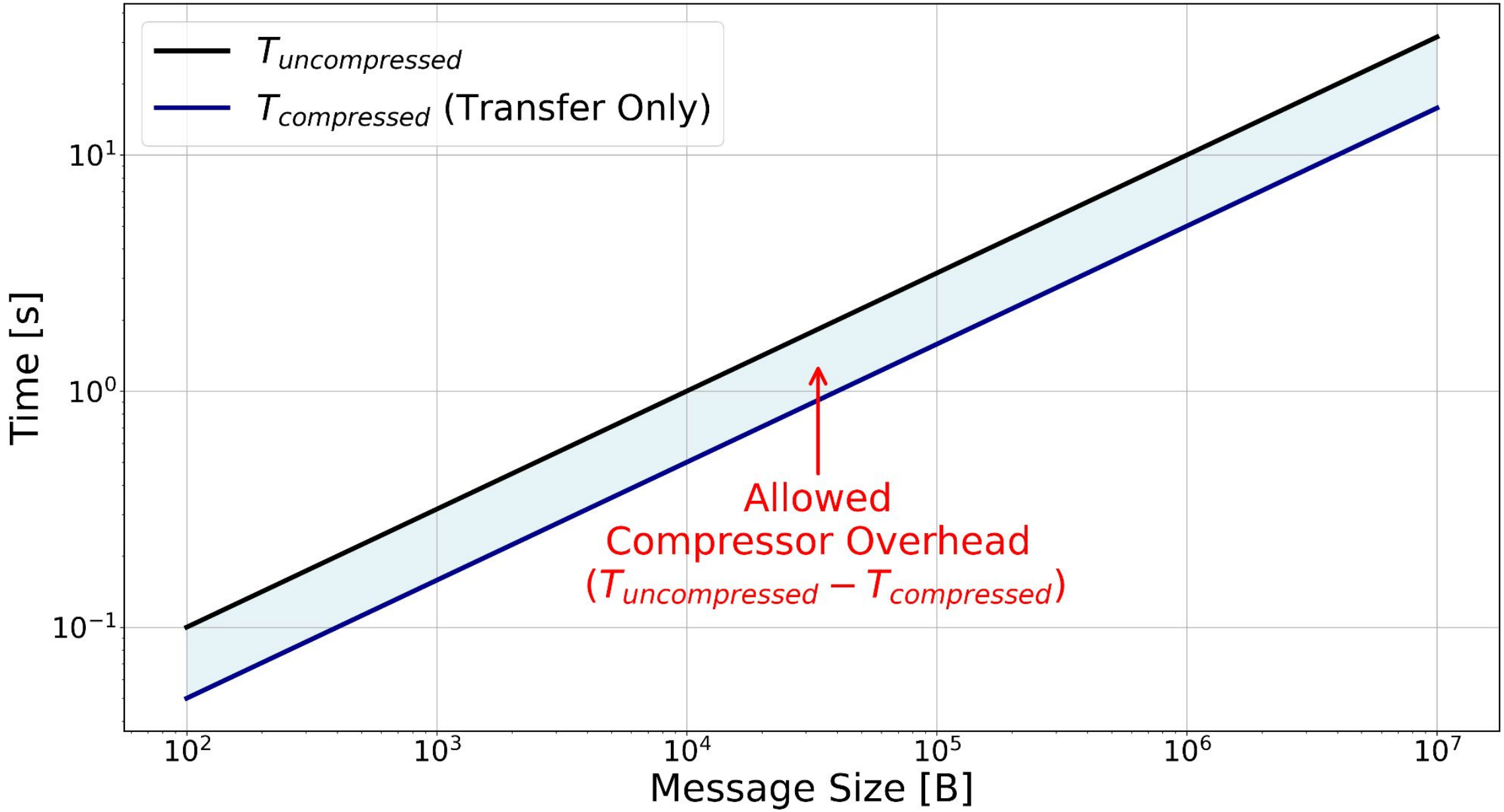


Figure: Intuitive derivation of a model to evaluate compressor overhead in the context on inline MPI message compression in GPU-accelerated applications

A Performance Model for Inline Compression in Point-to-Point exchanges

- Provide a **simple** way for a user to determine whether a **given compressor** could be used for in-line compression of MPI Point-to-Point exchanges

$$T_{\text{compressor}} < T_{\text{uncompressed}} - T_{\text{compressed}}$$

$$T_{\text{uncompressed}}(k, n) = \tau + \frac{k \cdot n}{\min(k \cdot BW_{\text{PP}}, BW_{\text{max}})}$$

$$T_{\text{compressed}}(k, n, r) = \tau + \frac{k \cdot r \cdot n}{\min(k \cdot BW_{\text{PP}}, BW_{\text{max}})}$$

$$t_{\text{comp}}(n, r) + t_{\text{decomp}}(n, r) < \frac{k \cdot n}{min(k \cdot BW_{\text{PP}}, BW_{\text{max}})} \cdot (1 - r)$$

Compressor Overhead Limit (COL)

A Performance Model for Inline Compression in Point-to-Point exchanges

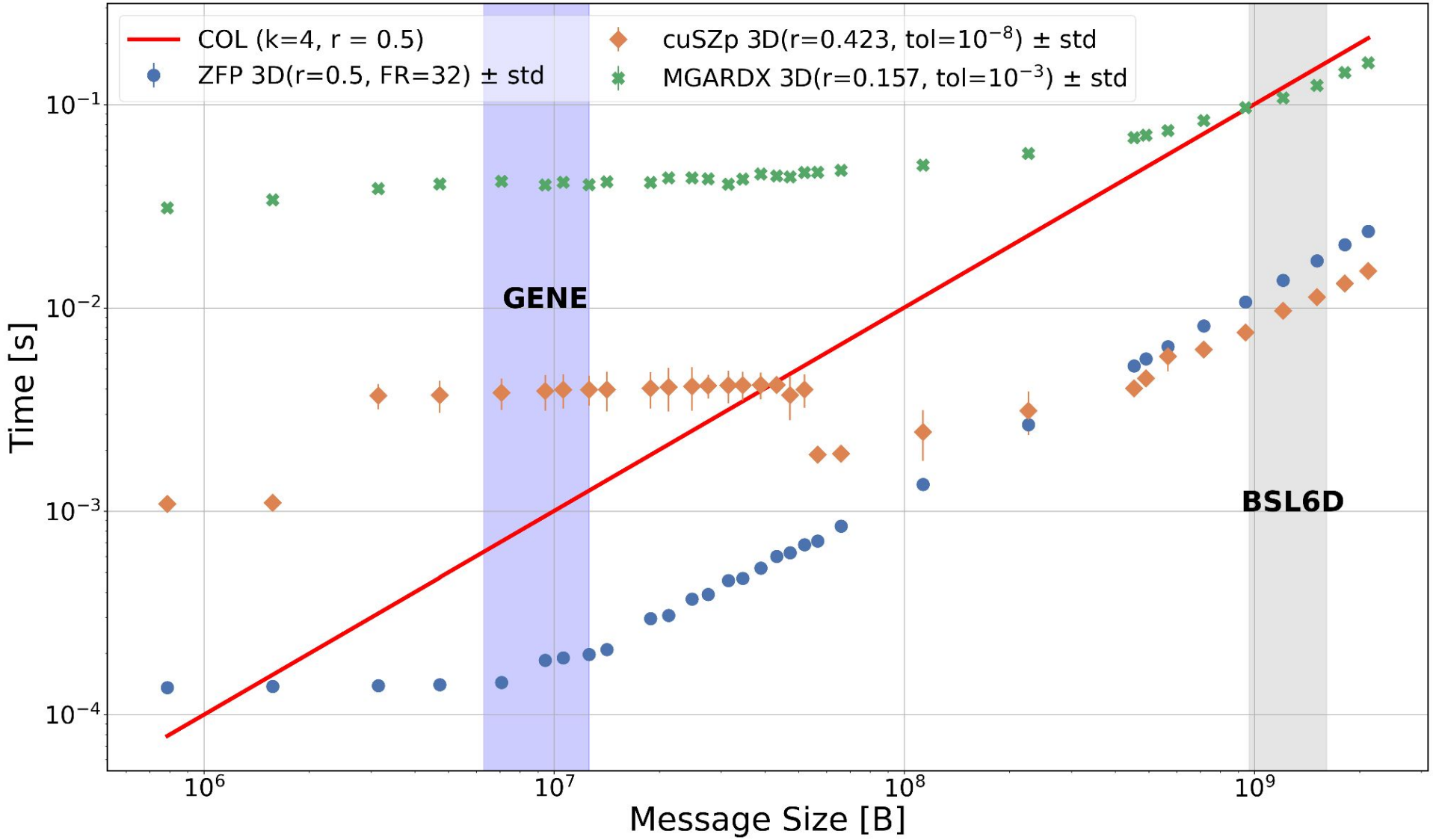


Figure: Evaluating state-of-the-art lossy compressors with the proposed Compressor Overhead Limit (COL)

A Performance Model for Inline Compression in Point-to-Point exchanges

$$T_{end_to end} = T_{\text{compressed}} + T_{\text{compressor}}$$

$$T_{end_to end} = \frac{1}{S} \cdot T_{\text{uncompressed}}(k, n)$$

$$T_{\text{compressor}}(n, r) < (\frac{1}{S} - 1) \cdot \tau + (\frac{1}{S} - r) \cdot \frac{k \cdot n}{\min(k \cdot BW_{\text{PP}}, BW_{\text{max}})}$$

A Performance Model for Inline Compression in Point-to-Point exchanges

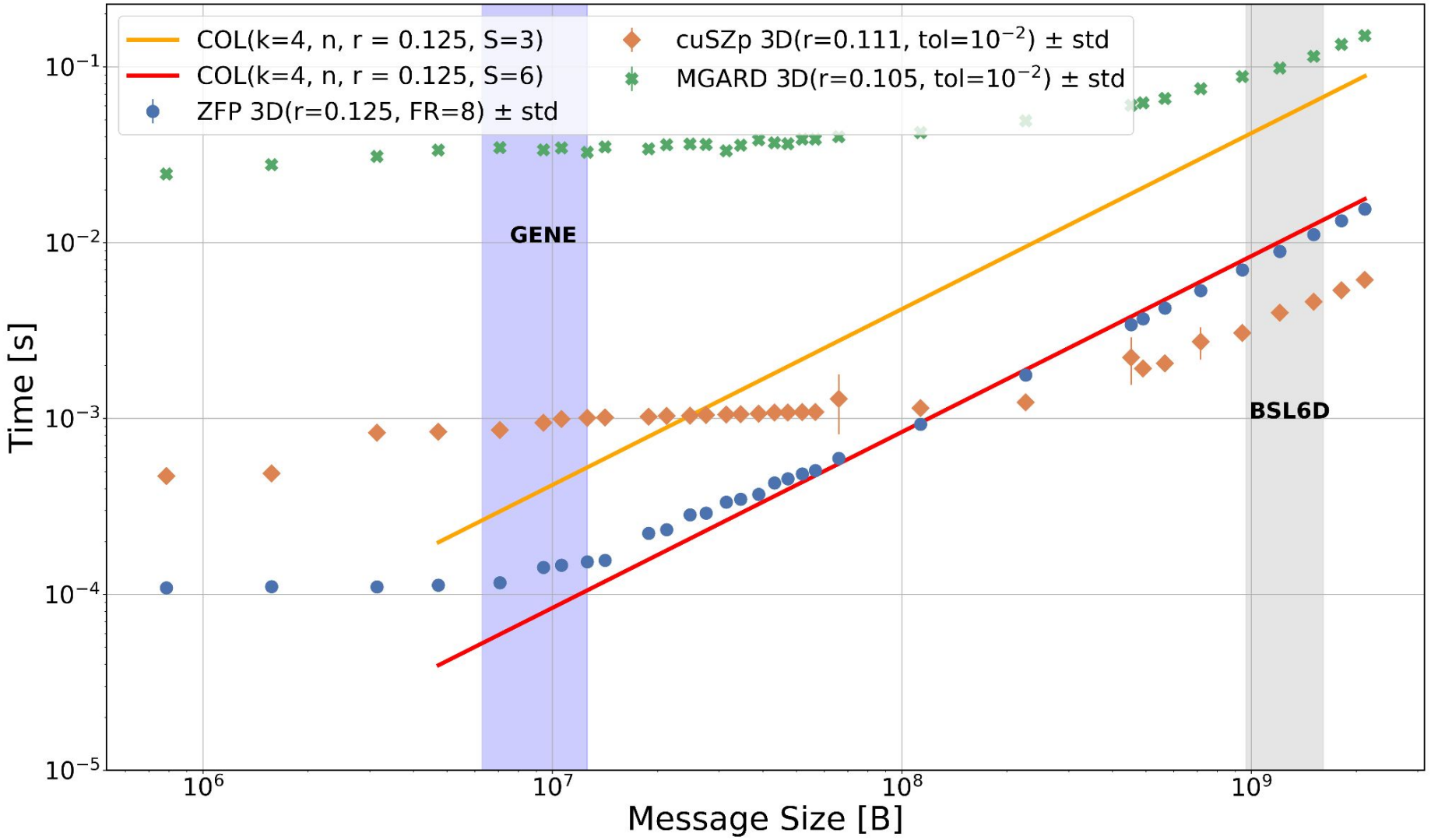
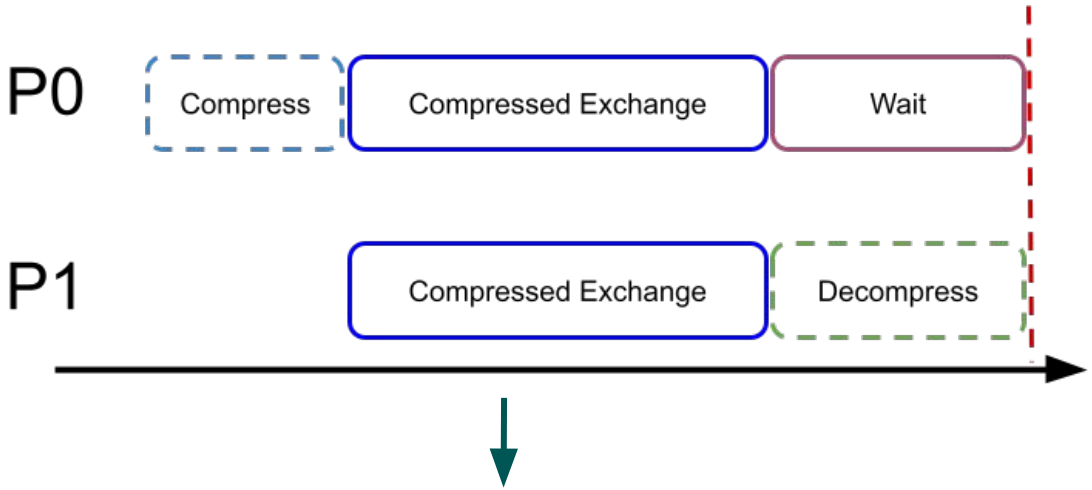


Figure: Evaluating state-of-the-art lossy compressors with the proposed Compressor Overhead Limit (COL)

Compressing Boundary Exchanges with ZFP

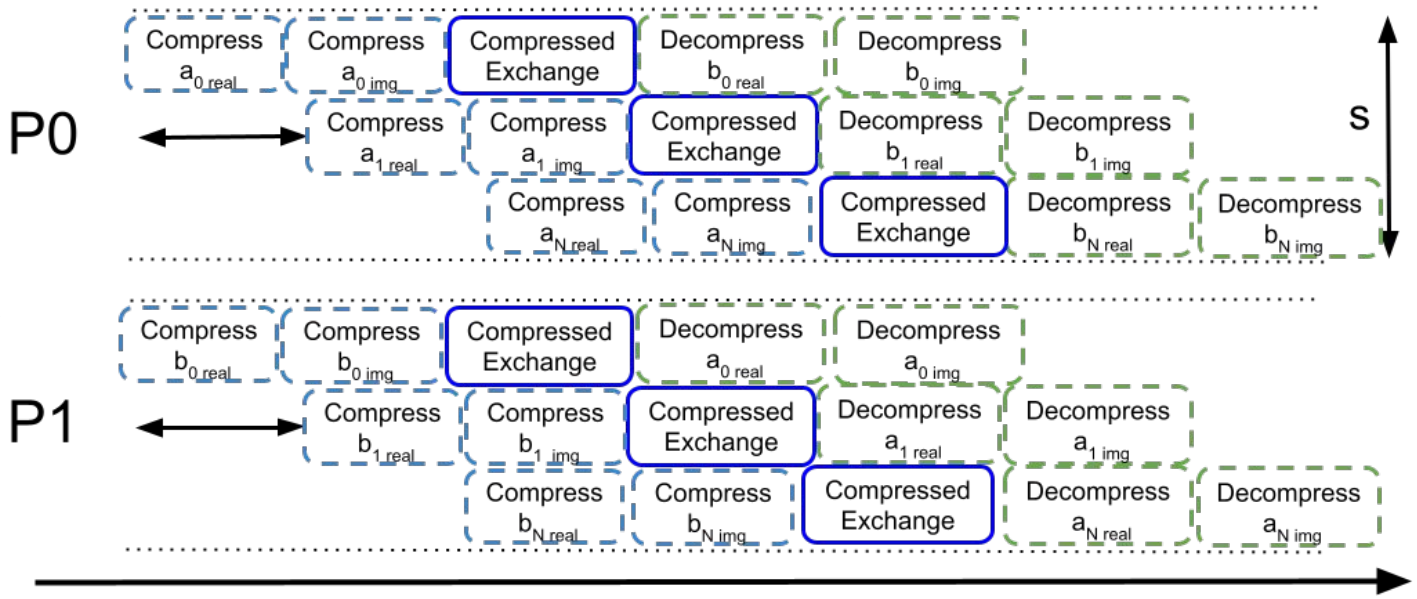
Compressing Boundary Exchanges with ZFP



Naive scheme is sufficient for BSL6D's halo exchange region

GENE relies on the complex data type (real and imaginary components)

MPI implementations have no knowledge of device execution queues (yet)



HIDING COMPRESSION COSTS IN GENE

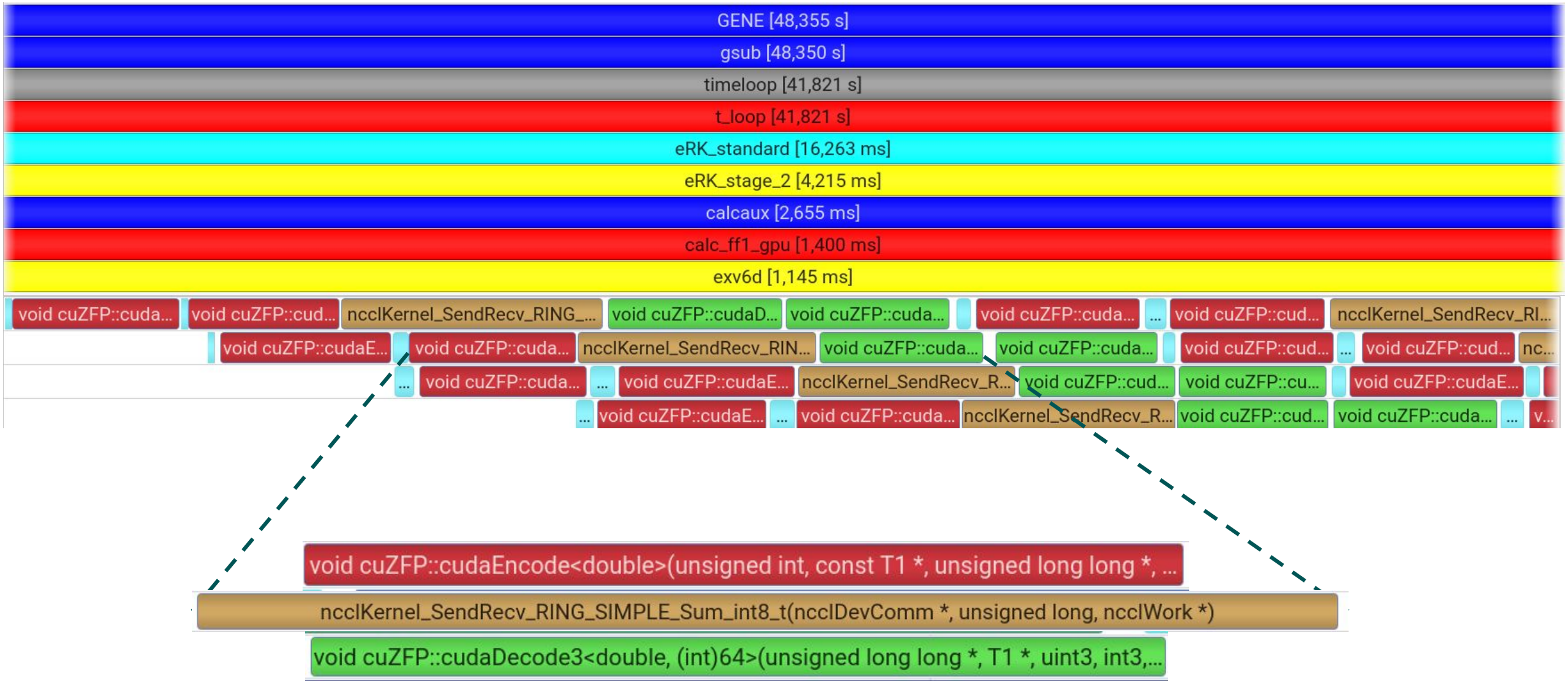


Figure: NSight Systems trace of an exv6d execution in GENE using multiple partitions, streams and duplicate communicators for compression-communication overlap

Application Results

BSL6D: ASSESSING COMPRESSED SIMULATION CORRECTNESS

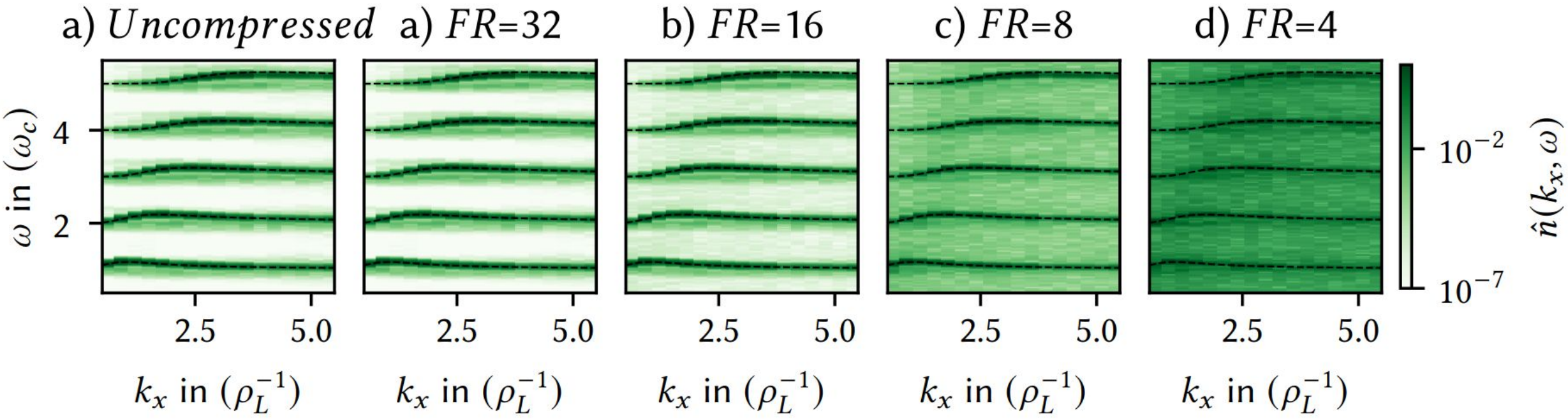


Figure: Neutralizing ion Bernstein wave dispersion relation solved with fixed-rate values FR=32,16,8,4 compared against a baseline run without compression. The dashed-black lines correspond to the analytical solution of the dispersion relation

BSL6D: ASSESSING COMPRESSED SIMULATION PERFORMANCE

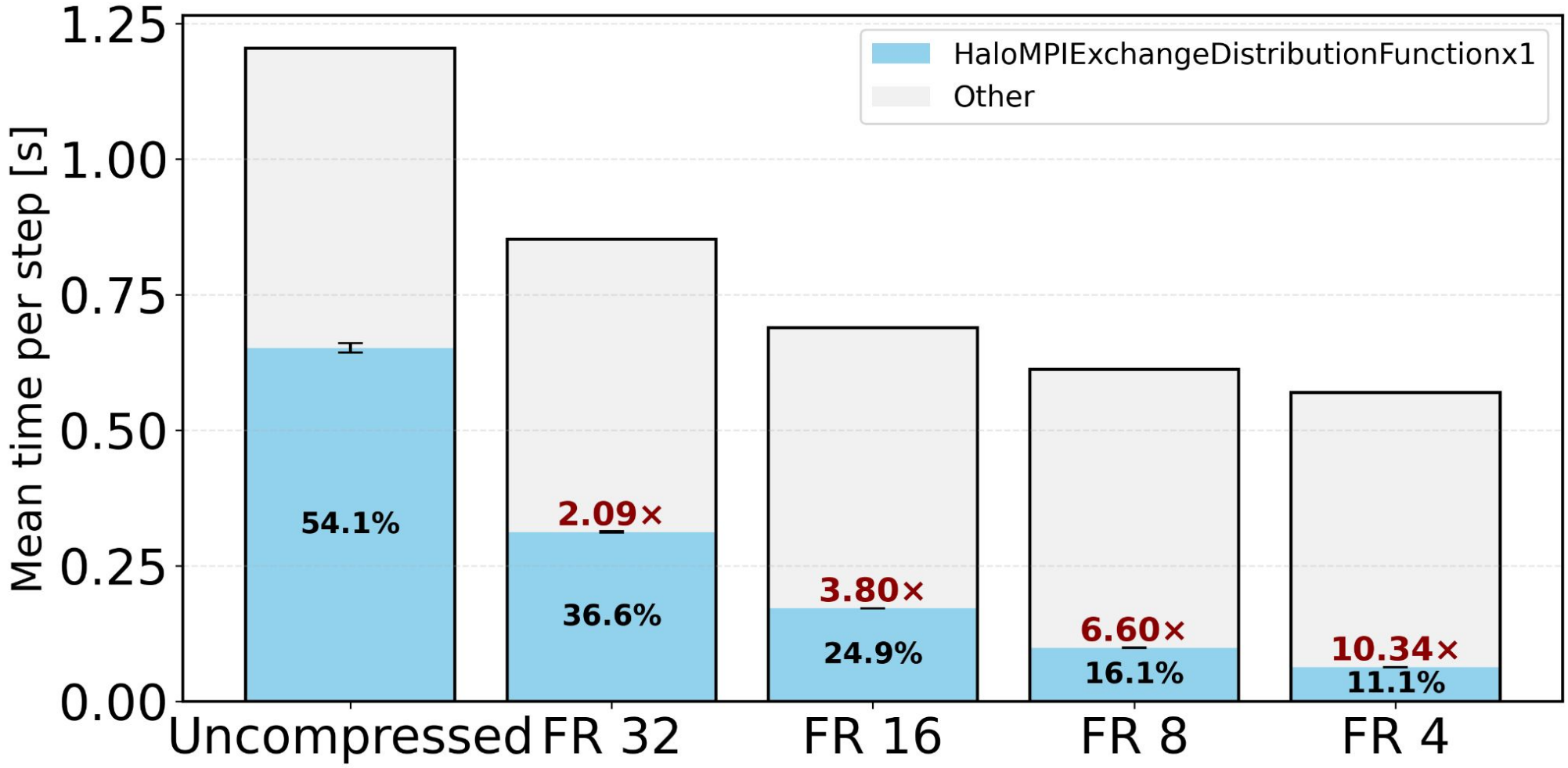


Figure: BSL6D time per iteration comparison for our test case between the uncompressed execution and the different ZFP fixed-rate values

GENE: ASSESSING COMPRESSED SIMULATION CORRECTNESS

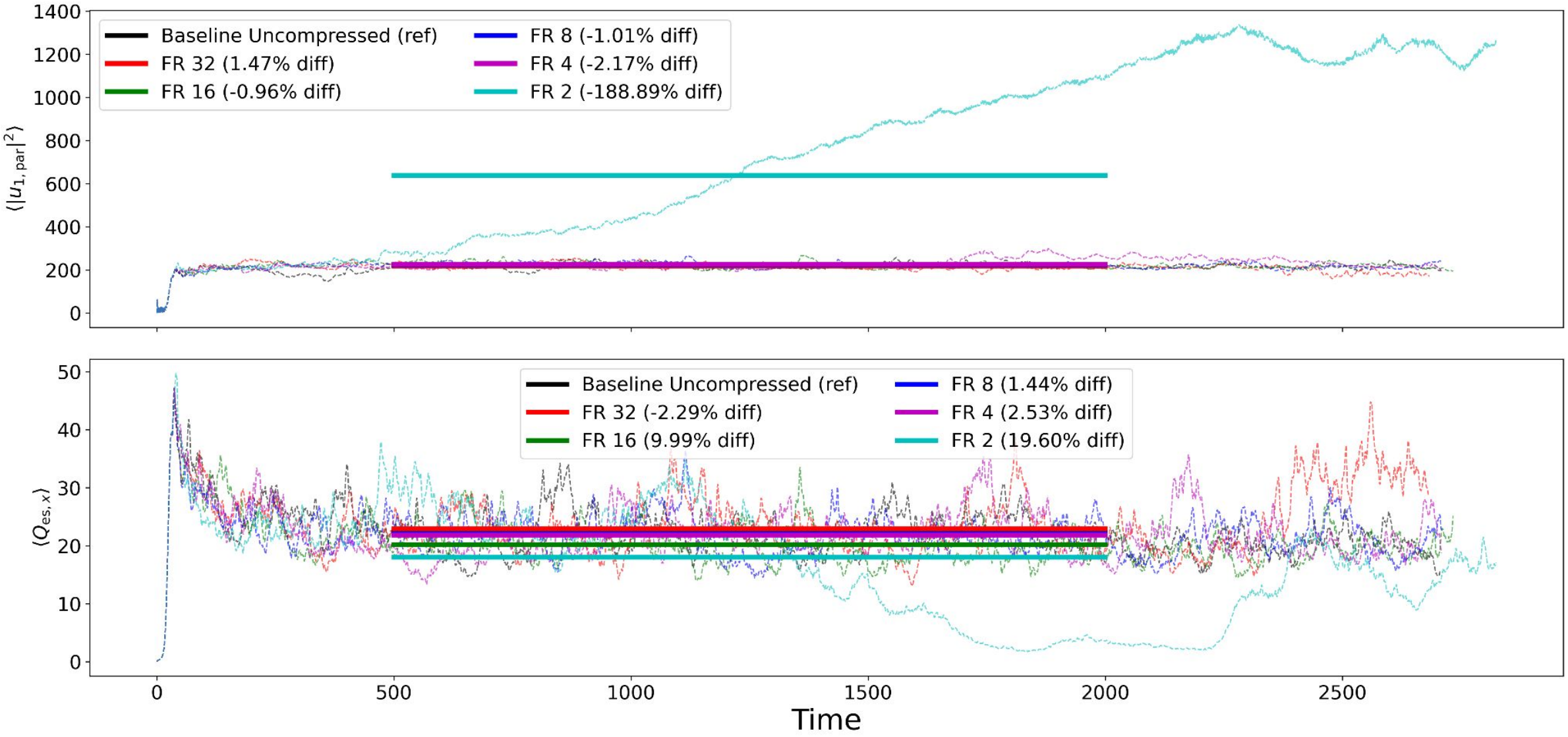


Figure: Simulation traces of spatially averaged electrostatic heat-flux, and parallel velocity moment with estimated means under different compression factors applied to the messages in the $v_{||}$ direction boundary exchange

GENE: ASSESSING COMPRESSED SIMULATION PERFORMANCE

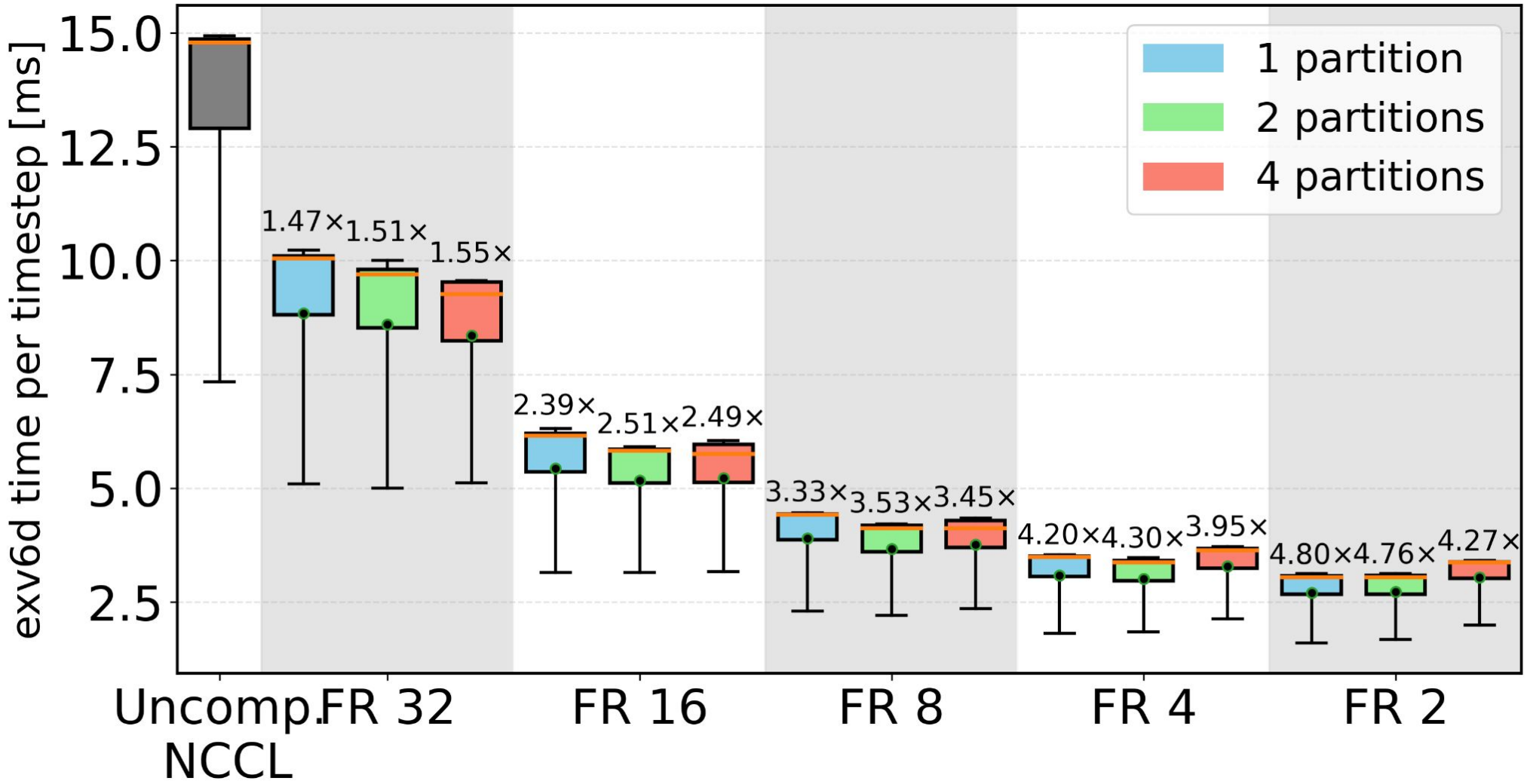


Figure: Distribution of GENE reported exv6d time per timestep across MPI ranks and their relation to different ZFP fixed-rate values and number of overlapping partitions in the proposed scheme

FINAL REMARKS

- **Inline lossy compression has the potential to alleviate bandwidth bottlenecks in point-to-point exchanges (large messages in kinetic applications)**
- **Coverage of impact over Quantities of Interest still limited and effort should be put into more robust science-verification framework**
- **Compressed communication libraries are currently in development providing already ‘transparent’ data reduction opportunities.**
- **What happens when we don’t/can’t have post-mortem baseline for analysis?**
  - This applies also for reduced-precision execution work within our project
  - Dynamic steering of compression factor during simulation

Thanks!

diego.jimenez@mpcdf.mpg.de

MPCDF MAX PLANCK COMPUTING & DATA FACILITY

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

ParTec MODULAR SUPERCOMPUTING

References

[1] Germaschewski, K., Allen, B., Dannert, T., Hrywniak, M., Donaghy, J., Merlo, G., ... & Bhattacharjee, A. (2021). Toward exascale whole-device modeling of fusion devices: Porting the GENE gyrokinetic microturbulence code to GPU. Physics of Plasmas, 28(6), 062501.

[2] Schild, N., Raeth, M., Eibl, S., Hallatschek, K., & Kormann, K. (2024). A performance portable implementation of the semi-Lagrangian algorithm in six dimensions. *Computer Physics Communications*, *295*, 108973

[3] Di, S., Liu, J., Zhao, K., Liang, X., Underwood, R., Zhang, Z., ... & Cappello, F. (2025). A survey on error-bounded lossy compression for scientific datasets. ACM computing surveys, 57(11), 1-38.

[4] Di, S., & Cappello, F. (2016, May). Fast error-bounded lossy HPC data compression with SZ. In 2016 ieee international parallel and distributed processing symposium (ipdps) (pp. 730-739). IEEE.

[5] Lindstrom, P. (2014). Fixed-rate compressed floating-point arrays. IEEE transactions on visualization and computer graphics, 20(12), 2674-2683.

[6] Ainsworth, M., Tugluk, O., Whitney, B., & Klasky, S. (2018). Multilevel techniques for compression and reduction of scientific data—the univariate case. Computing and Visualization in Science, 19(5), 65-76.