University of Stuttgart
Department of Mathematics

Checkpoint/Restart for Iterative Solvers utilising Lossy Compression

M. Altenbernd
&
D. Göddeke

7th Applied Mathematics Symposium Münster

November 11, 2019

ians
SimTech

CPR for Iterative Solvers utilising Lossy Compression

Motivation - Fault-tolerance

- More components at exascale and beyond $\rightarrow$ higher probability of failure
- Active debates to sacrifice reliability for energy efficiency
- Nightmare scenarios of MTBF $<$ 1 h

#cores	1	100	10 000	1 000 000
MTBF	5 years	18 days	4 hours	3 mins

Classical techniques

- Reliability in hardware (ECC protection etc.) too power-hungry
- Classical checkpoint/restart (CPR) too memory-intensive (and too slow)
- Triple modular redundancy too power-hungry, but: can be more energy-efficient and applicable for large fault rates

CPR for Iterative Solvers utilising Lossy Compression

Key objectives

- Efficient recovery from a partial data-loss, e.g. node-loss
- Low overhead in a fault-free scenario

Our approach

- Local recovery instead of global roll-back
- Lossy compression to reduce memory overhead
- In-memory checkpointing for efficiency

CPR for Iterative Solvers utilising Lossy Compression

Assumptions

- Problem is bandwidth-limited
- After a process failure a new process can be spawned and is able to
  1. Replace the old process in the communicator with a new one (ULFM[1])
  2. Catch up to the iterative solver locally, e.g. by
     - Message logging[2]
     - Checkpointing of persistent data (matrices, . . .)
     - . . .
  3. Receive the compressed backup from a remote processor

[1] W. Bland, A. Bouteiller, T. Herault, G. Bosilca, J. Dongarra, **Post-failure recovery of MPI communication capability: Design and rationale**. IJHPCA 27(3): 244-254, 2013

[2] C. Cantwell and A. Nielsen, **A Minimally Intrusive Low-Memory Approach to Resilience for Existing Transient Solvers**, Journal of Scientific Computing, 2019

CPR for Iterative Solvers utilising Lossy Compression

Assumptions

- Problem is bandwidth-limited
- After a process failure a new process can be spawned and is able to
  1. Replace the old process in the communicator with a new one (ULFM[1])
  2. Catch up to the iterative solver locally, e.g. by
     - Message logging[2]
     - Checkpointing of persistent data (matrices, . . .)
     - . . .
  3. Receive the compressed backup from a remote processor

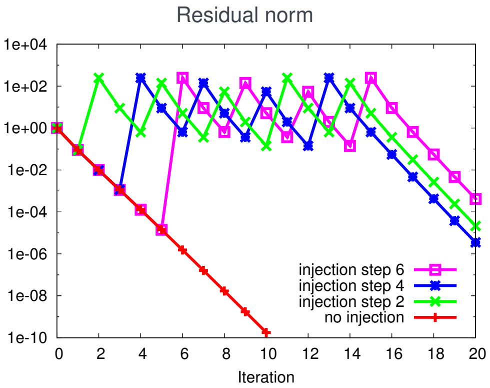
[1] W. Bland, A. Bouteiller, T. Herault, G. Bosilca, J. Dongarra, **Post-failure recovery of MPI communication capability: Design and rationale**. IJHPCA 27(3): 244-254, 2013

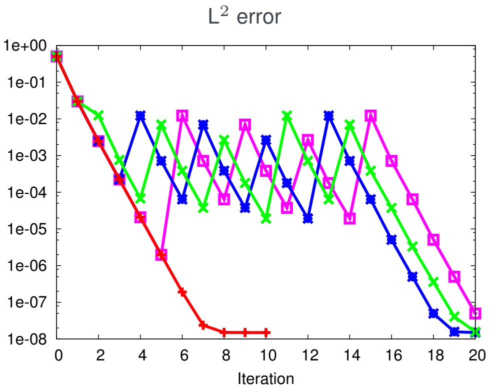
[2] C. Cantwell and A. Nielsen, **A Minimally Intrusive Low-Memory Approach to Resilience for Existing Transient Solvers**, Journal of Scientific Computing, 2019

Question

What happens if a part of the iterative data is lost?

Multigrid and local data-losses



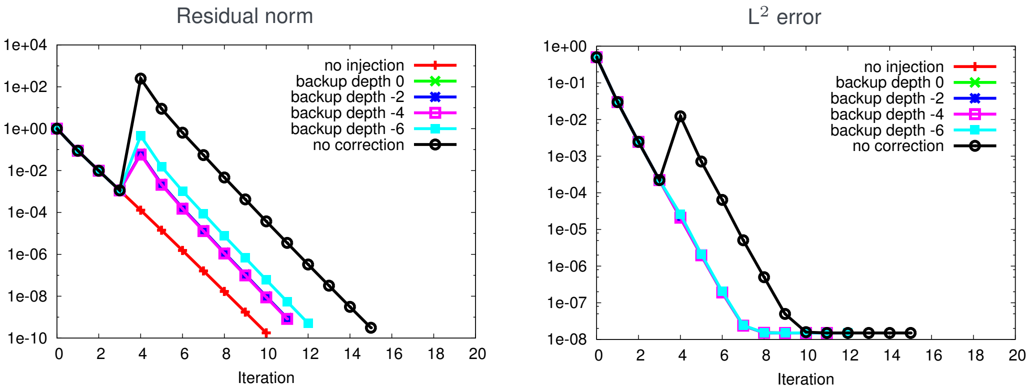


Observations

- A fault is comparable to a restart of the multigrid solver
- Multigrid converges always if the fault-rate is not to high
- Node-losses and Silent Data Corruptions show a similar behaviour

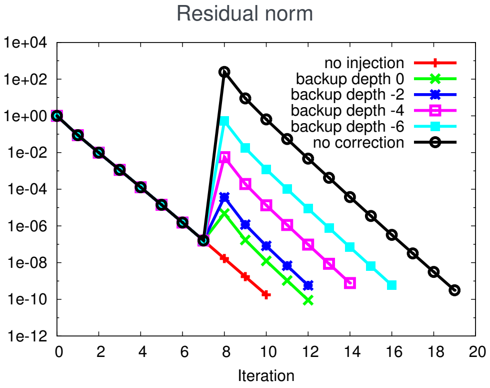
Multigrid compression

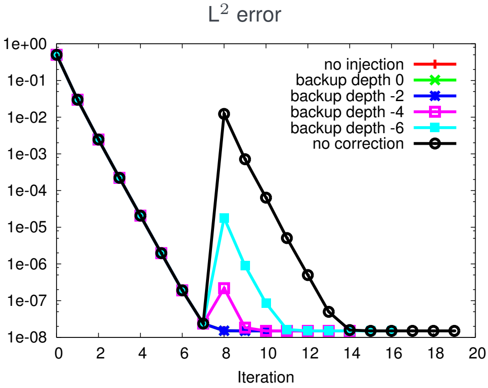
- Use multigrid transfer operators to compress checkpoint
- Data reduction in d dimensions: $\sim 2^d$ per backup depth (regular coarsening)
- Restore locally lost data with prolongated (decompressed) checkpoint



Multigrid compression

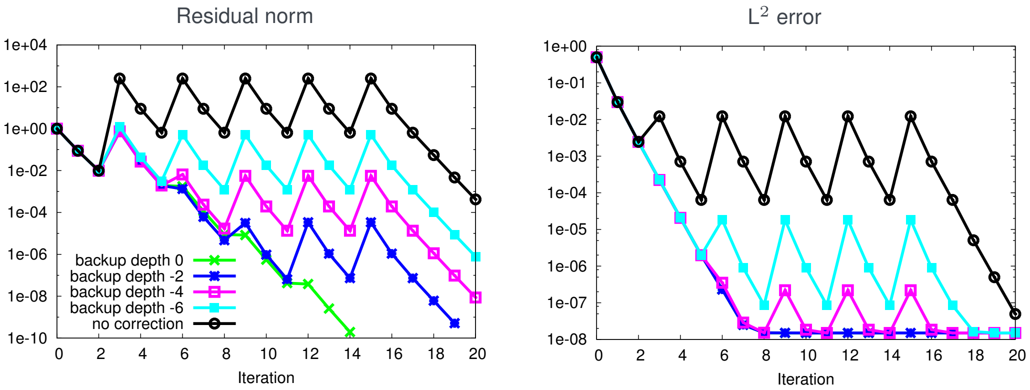
- Use multigrid transfer operators to compress checkpoint
- Data reduction in d dimensions: $\sim 2^d$ per backup depth (regular coarsening)
- Restore locally lost data with prolongated (decompressed) checkpoint





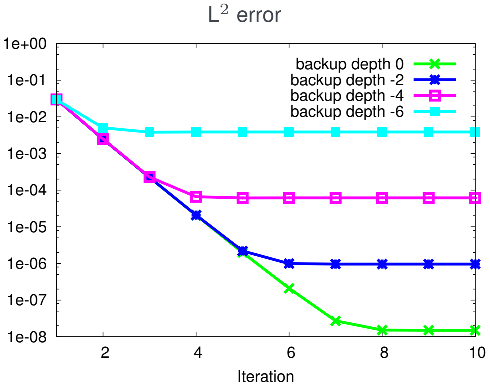
Multigrid compression

- Use multigrid transfer operators to compress checkpoint
- Data reduction in d dimensions: $\sim 2^d$ per backup depth (regular coarsening)
- Restore locally lost data with prolongated (decompressed) checkpoint



ians SimTech University of Stuttgart Germany

Multigrid compression



Problem

Discretisation error on backup level limits quality of repair on fine level

Improved restoration

Auxiliary problem (also independently developed by Huber et al.[3])

Extend faulty/lost indices $\mathcal{F} \subset \mathbb{N}$ which are owned by the processor, e.g. by using the connectivity pattern of operator $\mathbf{A}$ or an overlap, to $\mathcal{J}$ and solve

$$\begin{aligned} \mathbf{A}(\mathcal{J},\mathcal{J})\tilde{x}(\mathcal{J}) &= b(\mathcal{J}) && \text{in } \mathcal{F} \\ \tilde{x} &= x && \text{on } \mathcal{J}\backslash\mathcal{F} \end{aligned}$$

iteratively with initial guess $\tilde{x} = x_{cp}$ in $\mathcal{F}$.

Advantages

- Convergence behaviour can provably be restored
- Speed-up when using better checkpoints as initial guesses (iterative solver)
- Local problem: Possible to use a 'superman'[3] strategy for further speed-up

[3] M. Huber, B. Gmeiner, U. Rüde, B. Wohlmuth, **Resilience for Massively Parallel Multigrid Solvers**, SIAM Journal on Scientific Computing, 2016

Observations

- Multigrid can be used for lossy compression
- Rate of compression is adaptive and predictable but not its accuracy
- Compression rate must be lowered during the iterative procedure:
  There is no naive way to do this adaptively
- Eventually an auxiliary problem has to be solved to ensure convergence
- The decompressed data is a good initial guess for this auxiliary problem

But

Multigrid is good preconditioner, but rarely a standalone solver

D. Göddeke, M.A., D. Ribbrock, **Fault-tolerant finite-element multigrid algorithms with hierarchically compressed asynchronous checkpointing**, Parallel Computing, 2015

New approach

Application-oriented solvers

- CG, BiCGStab, GMRES, . . .
- Nested solvers, Inner-outer approaches, Newton-like methods, . . .

Evaluating impact of

- Various (lossy) compression techniques
  - Multigrid compression
  - SZ compression
- Different restoration methods
  - Local restoration based on compressed checkpoint
  - Global roll-back to compressed checkpoint
  - Improved restoration by solving auxiliary problem
- Variable checkpoint frequencies

SZ compression

How it works

- Save initial point value (with reduced accuracy): Unpredictable data
- Predict next point based on previously processed points via an interpolation based on n layers
- Compare predicted value with real value and improve it through a *Huffman*-like coding
- If still not ‘close enough’ data is stored as unpredictable and compressed via binary-representation analysis

Advantages and disadvantages

- Adaptive controllable compression accuracy (via parameter)
- More computational overhead than multigrid compression
- Lower compression rate $\rightarrow$ higher computation time

D. Tao, S. Di, Z. Chen and F. Cappello, **Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization**, Computing Research Repository, 2017

SZ compression

- Different error control strategies available:
  - Absolute, relative, ...
  - Global, point-wise, ...
- Point-wise relative (PW_REL) error bound:

$$\frac{|\tilde{x}_j - x_j|}{|x_j|} \leq \texttt{eb}_{\texttt{PW\_REL}}, \qquad \forall j = 1, \ldots, N$$

where $x \in \mathbb{R}^N$ is the input data and $\tilde{x} \in \mathbb{R}^N$ its decompressed counterpart. This implies that

$$\|\tilde{x} - x\|_2 \leq \texttt{eb}_{\texttt{PW\_REL}} \, \|x\|_2.$$

- Different strategies for choosing $\texttt{eb}_{\texttt{PW\_REL}}$:
  1. Fixed over the iterative solve
  2. Adaptive (coupled to readily available quantities)

Adaptive SZ compression

Residual error after decompression:

$$\|A\tilde{x}^{(i)} - b\|_2 \leq \|Ax^{(i)} - b\|_2 + \|A(\tilde{x}^{(i)} - x^{(i)})\|_2$$

with

$$\|A(\tilde{x}^{(i)} - x^{(i)})\|_2 \leq \sup_{x \in \mathbb{R}^N} \frac{\|Ax\|_2}{\|x\|_2} \|\tilde{x}^{(i)} - x^{(i)}\|_2 \overset{\text{PW\_REL}}{\leq} \sup_{x \in \mathbb{R}^N} \frac{\|Ax\|_2}{\|x\|_2} \mathtt{eb}_{\mathtt{PW\_REL}} \; \|x^{(i)}\|_2 .$$

Omitting the supremum and just considering the current iteration vector $x^{(i)}$ yields

$$\|A(\tilde{x}^{(i)} - x^{(i)})\|_2 \approx \frac{\|Ax^{(i)}\|_2}{\|x^{(i)}\|_2} \mathtt{eb}_{\mathtt{PW\_REL}} \; \|x^{(i)}\|_2 = \|Ax^{(i)}\|_2 \, \mathtt{eb}_{\mathtt{PW\_REL}} \xrightarrow{i \to \infty} \|b\|_2 \, \mathtt{eb}_{\mathtt{PW\_REL}} \, .$$

Adaptive SZ compression

This yields

$$\|A\tilde{x}^{(i)} - b\|_2 \approx \|Ax^{(i)} - b\|_2 + \|b\|_2 \, \mathtt{eb}_{\mathtt{PW\_REL}} \,.$$

The introduced compression error will be of similar order of magnitude as the iterative solver residual when

$$\mathtt{eb}_{\mathtt{PW\_REL}} = \frac{\|Ax^{(i)} - b\|_2}{\|b\|_2} \, \mathtt{tol}_{\mathtt{aSZ}} \,.$$

- $\mathtt{tol}_{\mathtt{aSZ}}$ is a tuning parameter
- Lower $\mathtt{tol}_{\mathtt{aSZ}} \rightarrow$ lower $\|A(\tilde{x}^{(i)} - x^{(i)})\|_2$ and $\|\tilde{x}^{(i)} - x^{(i)}\|_2$
  - $\rightarrow$ Diminishing compression error influence
  - $\rightarrow$ Lower compression rate **and** probably an increased compression time

Numerical tests

- Anisotropic diffusion in 2D with Dirichlet-boundary condition:

$$-\nabla \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0.01 \end{pmatrix} \nabla u = b$$

- Exact solution $u(x, y) = \sin(\pi x^2)\sin(\pi y^2)$
- Linear finite elements on partitioned grid (52 ranks, overlapping Schwarz)
- ~$310\,000$ degrees of freedom per rank
- Preconditioner: Algebraic multigrid (7 levels, one V-cycle)
- Solver: Conjugate gradient $\rightarrow$ **115 iterations** until convergence
- Auxiliary solver reduction to absolute residuum norm of

$$\texttt{locale\_def\_norm\_at\_backup\_time} * 10^{-(\texttt{age\_of\_backup}+1)}$$

- Data-loss and recovery on rank $\{0, 21, 37\}$ after iteration $\{10, 40, 75, 110\}$

Compression/recovery strategies

Compression techniques

- **Zero**
  No backup $\rightarrow$ 'Zero recovery'
- **MG**
  2 levels $\rightarrow$ Compression 14-16x
- **SZ**_$\texttt{tol}_{\texttt{SZ}}$
  Fixed error bound

$$\texttt{eb}_{\texttt{PW\_REL}} = \texttt{tol}_{\texttt{SZ}}$$

- **aSZ**_$\texttt{tol}_{\texttt{aSZ}}$
  Adaptive error bound

$$\texttt{eb}_{\texttt{PW\_REL}} = \frac{\|Ax^{(i)} - b\|_2}{\|b\|_2} \texttt{tol}_{\texttt{aSZ}}$$

Recovery approaches

- **Global** rollback
  All data is replaced by backup data
- **Local** recovery
  Lost-data is re-initialising from backup
- **Improved** recovery
  Local recovery is improved by an auxiliary solve

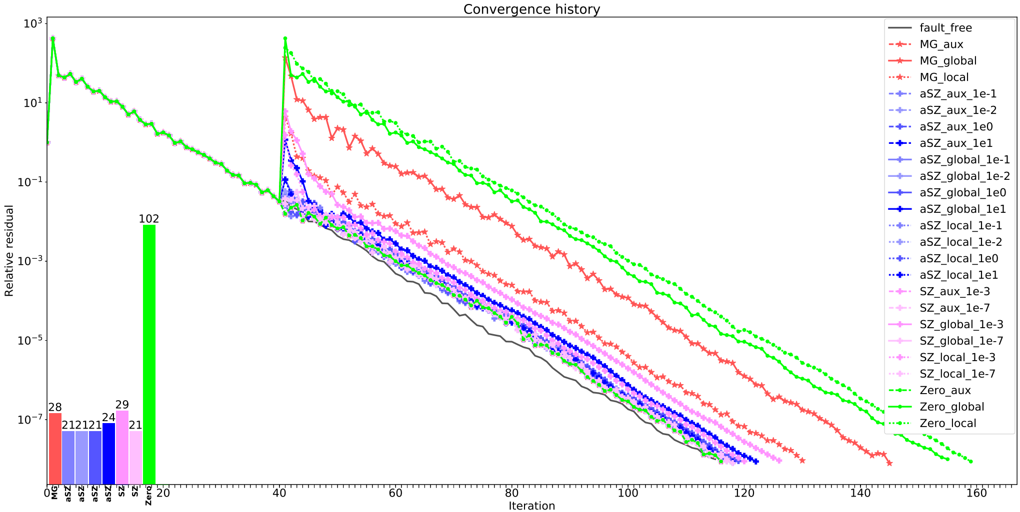
Numerical results - Iterations

	Zero	MG	SZ_1e-7	SZ_1e-3	aSZ_1e-2	aSZ_1e-1	aSZ_1e0	aSZ_1e1
Average								
global	200.00	200.00	154.00	193.00	116.00	115.00	119.00	120.00
local	199.33	194.67	134.33	173.00	114.33	115.67	115.67	114.67
improved	113.00	113.00	113.00	113.00	113.00	113.00	113.00	113.00
iterations	117.33	60.00	30.92	56.08	19.67	19.83	20.33	26.33
Fault-iteration: 10								
global	124.00	119.00	116.00	117.0	116.00	117.00	119.00	126.00
local	124.33	116.67	116.00	116.0	116.00	116.00	116.67	119.00
improved	115.33	115.00	115.00	115.0	115.00	115.00	115.00	115.00
iterations	46.33	17.67	18.67	19.0	18.67	18.67	19.00	27.67
Fault-iteration: 40								
global	154.00	144.0	117.00	125.00	117.00	117.00	118.00	121.00
local	148.33	129.0	115.67	118.33	115.67	116.00	115.67	116.67
improved	115.00	115.00	115.00	115.00	115.00	115.00	115.00	115.00
iterations	91.67	29.00	19.67	26.33	19.67	19.67	20.00	22.00
Fault-iteration: 75								
global	189.00	179.00	123.00	158.00	116.00	116.00	118.00	121.00
local	183.33	163.00	117.00	140.33	115.67	115.67	115.67	116.33
improved	115.00	115.00	115.00	115.00	115.00	115.00	115.00	115.00
iterations	141.33	75.00	28.00	72.33	17.33	18.00	18.33	27.33
Fault-iteration: 110								
global	200.00	200.00	154.00	193.00	116.00	115.00	119.00	120.00
local	199.33	194.67	134.33	173.00	114.33	115.67	115.67	114.67
improved	113.00	113.00	113.00	113.00	113.00	113.00	113.00	113.00
iterations	190.00	118.33	57.33	106.67	23.00	23.00	24.00	28.33

Numerical results - Iterations

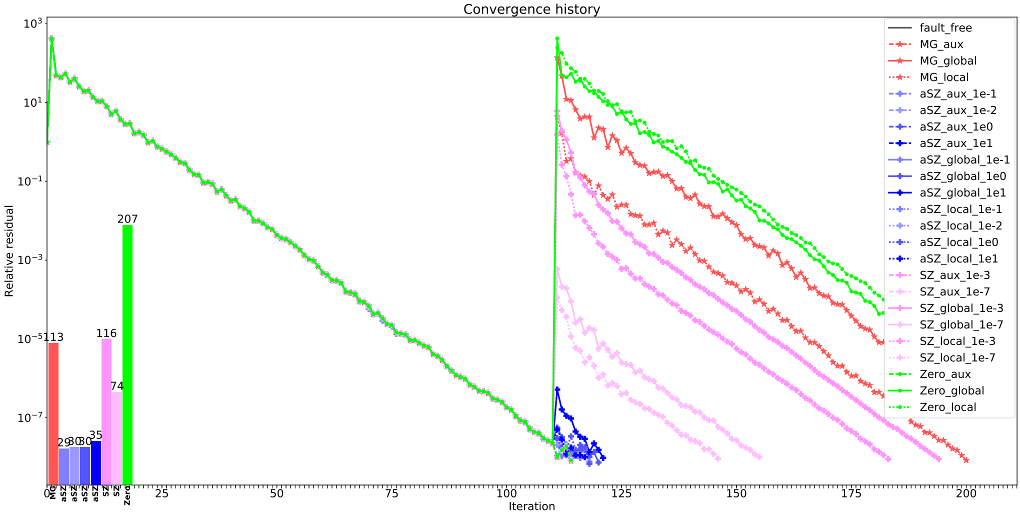
	Zero	MG	SZ_1e-7	SZ_1e-3	aSZ_1e-2	aSZ_1e-1	aSZ_1e0	aSZ_1e1
Average								
global	200.00	200.00	154.00	193.00	116.00	115.00	119.00	120.00
local	199.33	194.67	134.33	173.00	114.33	115.67	115.67	114.67
improved	113.00	113.00	113.00	113.00	113.00	113.00	113.00	113.00
iterations	117.33	60.00	30.92	56.08	19.67	19.83	20.33	26.33
Fault-iteration: 10								
global	124.00	119.00	116.00	117.0	116.00	117.00	119.00	126.00
local	124.33	116.67	116.00	116.0	116.00	116.00	116.67	119.00
improved	115.33	115.00	115.00	115.0	115.00	115.00	115.00	115.00
iterations	46.33	17.67	18.67	19.0	18.67	18.67	19.00	27.67
Fault-iteration: 40								
global	154.00	144.0	117.00	125.00	117.00	117.00	118.00	121.00
local	148.33	129.0	115.67	118.33	115.67	116.00	115.67	116.67
improved	115.00	115.00	115.00	115.00	115.00	115.00	115.00	115.00
iterations	91.67	29.00	19.67	26.33	19.67	19.67	20.00	22.00
Fault-iteration: 75								
global	189.00	179.00	123.00	158.00	116.00	116.00	118.00	121.00
local	183.33	163.00	117.00	140.33	115.67	115.67	115.67	116.33
improved	115.00	115.00	115.00	115.00	115.00	115.00	115.00	115.00
iterations	141.33	75.00	28.00	72.33	17.33	18.00	18.33	27.33
Fault-iteration: 110								
global	200.00	200.00	154.00	193.00	116.00	115.00	119.00	120.00
local	199.33	194.67	134.33	173.00	114.33	115.67	115.67	114.67
improved	113.00	113.00	113.00	113.00	113.00	113.00	113.00	113.00
iterations	190.00	118.33	57.33	106.67	23.00	23.00	24.00	28.33

Numerical results - Convergence behaviour



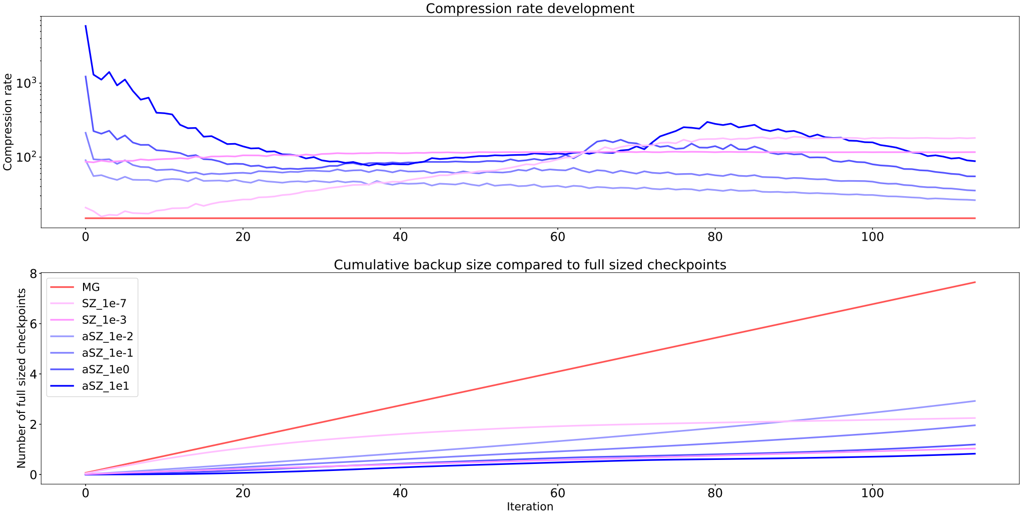
Data-loss after iteration 40 on rank 21.

Numerical results - Convergence behaviour



Data-loss after iteration 1190 on rank 21.

Numerical results - Compression rates



Note: Average over all 52 processors.

Performance analysis

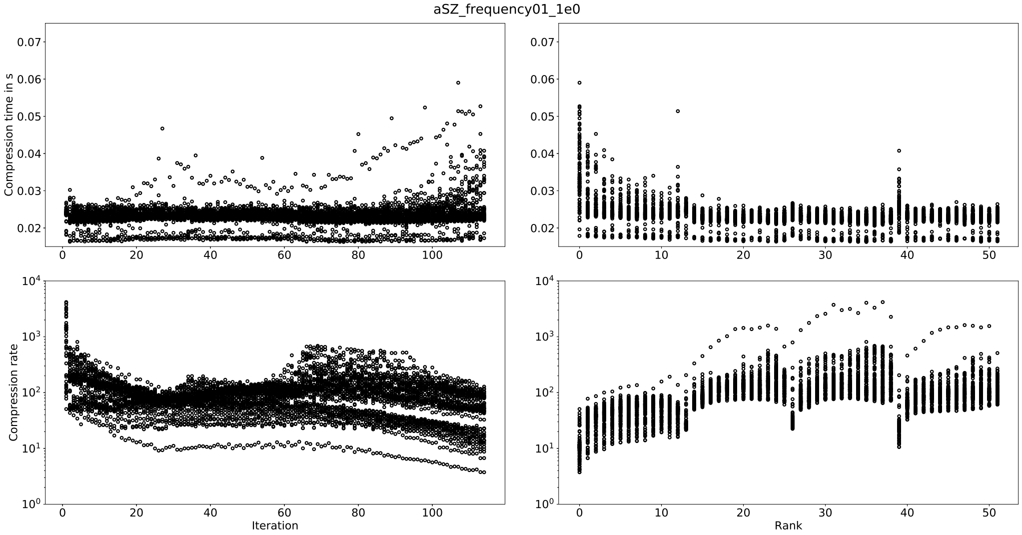
Settings

- Same problem as described before but fault-free
- Two nodes with Intel(R) Xeon(R) Gold 5120 CPU (2 $\times$ 14 cores):
  - Base frequency: 2.20 GHz
  - Turbo frequency: 3.20 GHz
  - L3 Cache: 19.15 MB
- 2 $\times$ 384 GB RAM
- Hyper-threading, but using only 52 threads

Baseline

- 16 184 529 degrees of freedom $\Rightarrow$ ca. 310 000 per core
- 115 iterations until convergence
- One iteration takes on average 7 seconds

Numerical results - aSZ compression



Numerical results - Overall solver runtime

in seconds

Fault-free

Zero	MG	SZ_1e-7	SZ_1e-3	aSZ_1e-2	aSZ_1e-1	aSZ_1e0	aSZ_1e1
779.14	783.99	780.69	782.13	782.21	784.26	780.60	783.23

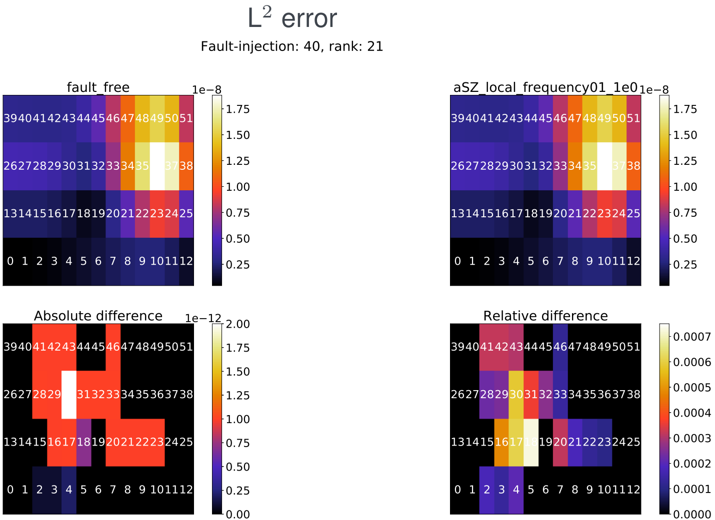
Average

Data-loss and recovery on rank $\{0, 21, 37\}$ after iteration $\{10, 40, 75, 110\}$

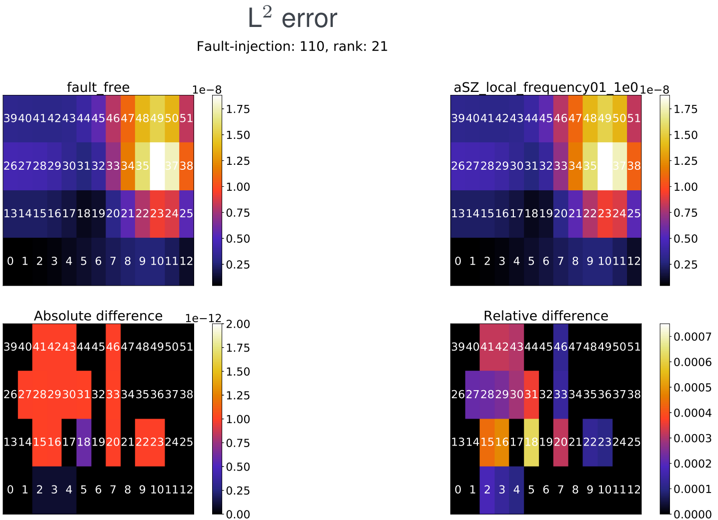
	Zero	MG	SZ_1e-7	SZ_1e-3	aSZ_1e-2	aSZ_1e-1	aSZ_1e0	aSZ_1e1
global	1149.73	1117.37	886.32	1026.24	812.46	809.76	825.83	849.50
local	1129.41	1050.12	841.48	951.49	806.14	807.52	808.17	813.28
improved	968.44	890.62	844.41	880.95	828.67	828.37	829.64	838.33
iterations	117.33	60.00	30.92	56.08	19.67	19.83	20.33	26.33

Caution: Iterations still indicates an iteration count and not a time measurement.

Numerical results - Error



Numerical results - Error



Numerical results - Lower frequency

	aSZ_1e-1	aSZ_1e0	aSZ_1e1
Fault-iteration: 108			
global	119.0	121.0	124.0
local	118.0	115.0	118.0
improved	114.0	114.0	114.0
iterations	64.0	65.0	61.0
Fault-iteration: 109			
global	118.0	117.0	120.0
local	117.0	116.0	119.0
improved	112.0	127.0	112.0
iterations	80.0	86.0	80.0

	aSZ_1e-1	aSZ_1e0	aSZ_1e1
Fault-iteration: 110			
global	116.0	118.0	121.0
local	117.0	117.0	117.0
improved	111.0	111.0	111.0
iterations	46.0	49.0	48.0
Fault-iteration: 111			
global	116.0	120.0	121.0
local	116.0	118.0	115.0
improved	111.0	111.0	111.0
iterations	45.0	53.0	46.0

	aSZ_1e-1	aSZ_1e0	aSZ_1e1
Average			
global	117.25	119.00	121.50
local	117.00	116.50	117.25
improved	112.00	115.75	112.00
iterations	58.75	41.75	58.75

Backup after every 4th iteration.

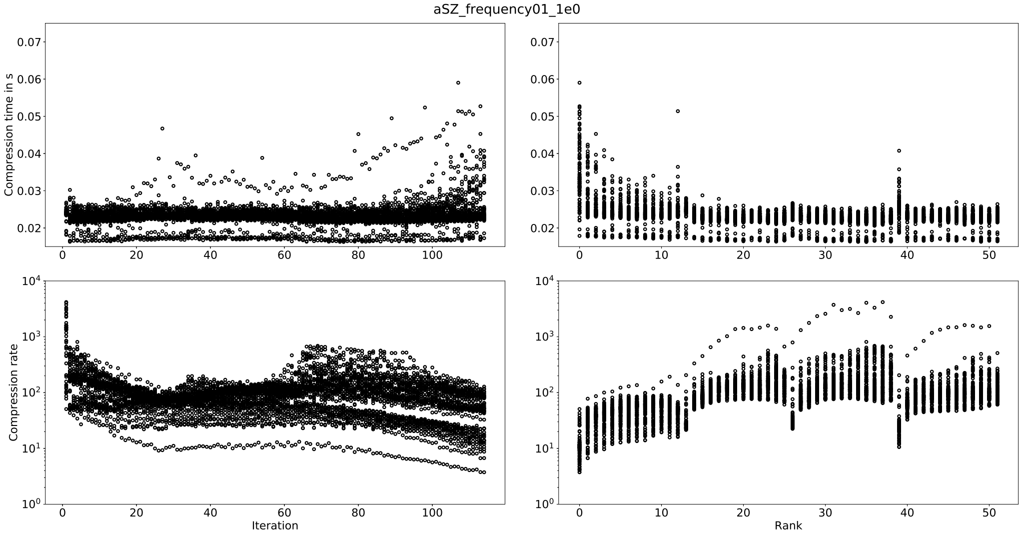
Summary

- We have combined local recovery with lossy compression
- The obtained method is able to recover using two different approaches
  1. A simple local restoration with possibly some additional global iterations
  2. A local auxiliary problem which can be speed-up by a ‘superman’ strategy
- Compression target is coupled to local defect norm
  - Early on a high compression rate can be achieved
  - Backup quality is increased towards the end
- Overhead in the fault-free scenario is minimal
- Communication overhead is significantly reduced compared to full sized checkpoints
  - Can be further reduced by using lower checkpoint frequencies
  - Asynchronous checkpointing could dispense the communication

Acknowledgements

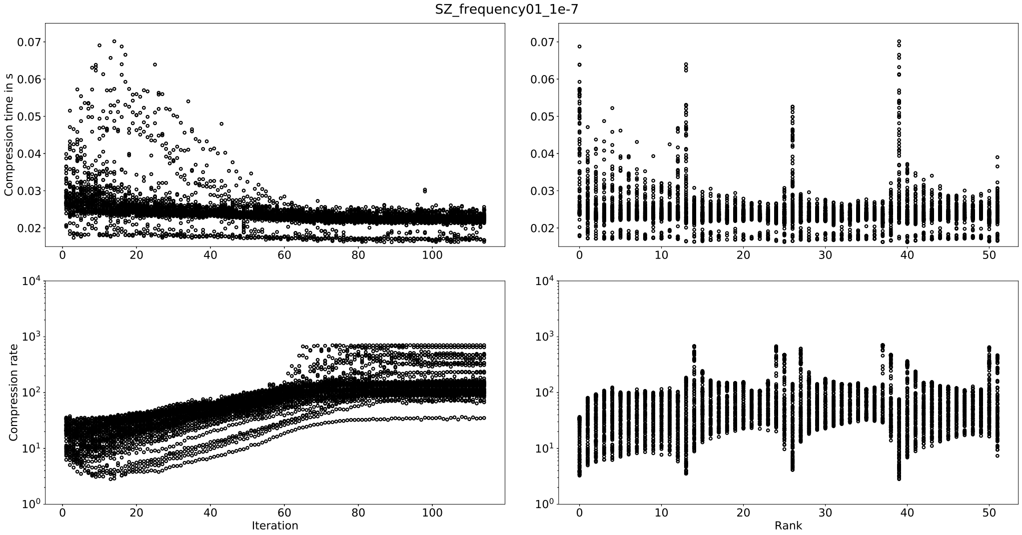
DFG Priority Program 1648 ’Software for Exascale Computing’, grant GO 1758/2-2

Compression behaviour: aSZ vs. SZ

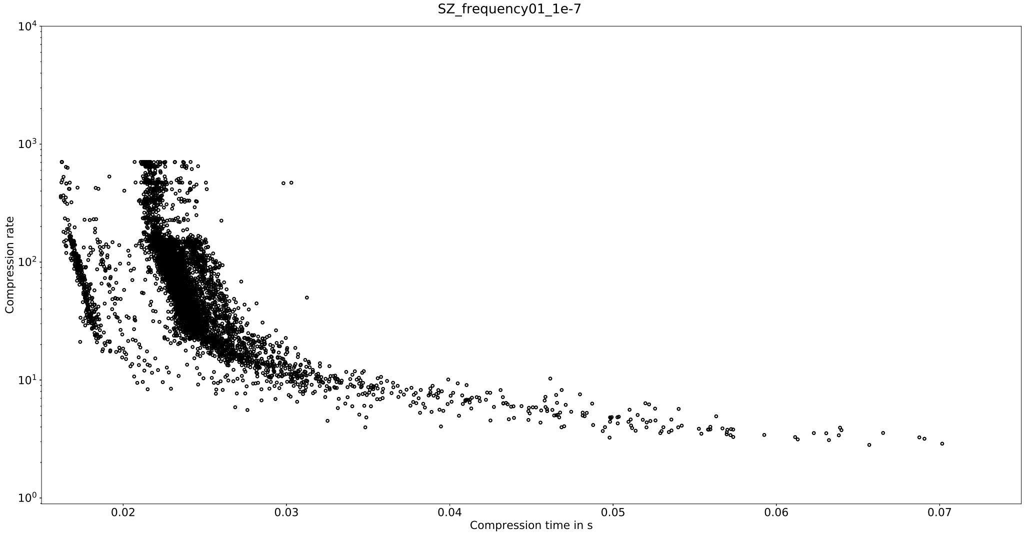


ians SimTech University of Stuttgart Germany

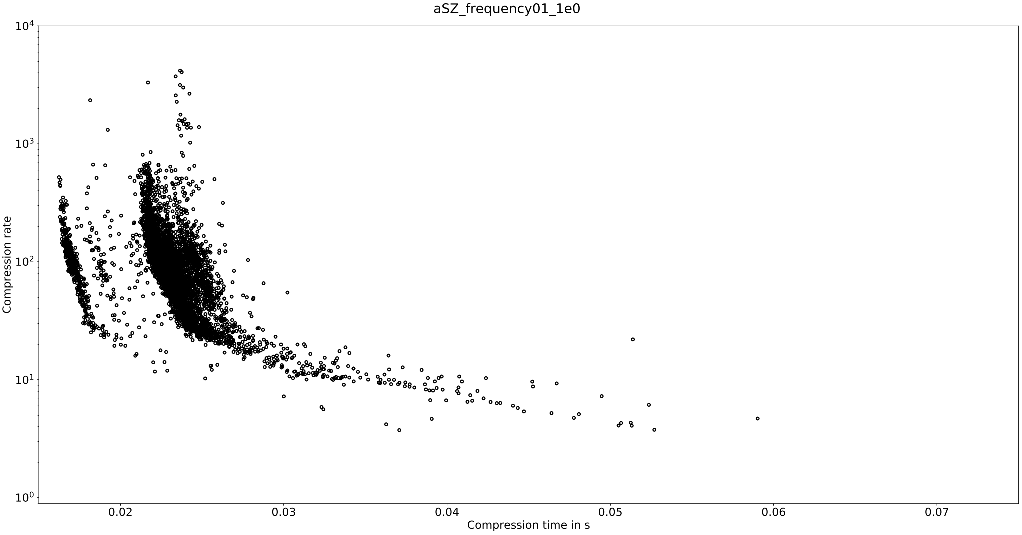
Compression behaviour: aSZ vs. SZ



Compression time vs. compression rate



Compression time vs. compression rate



Numerical results - Overall solver runtime

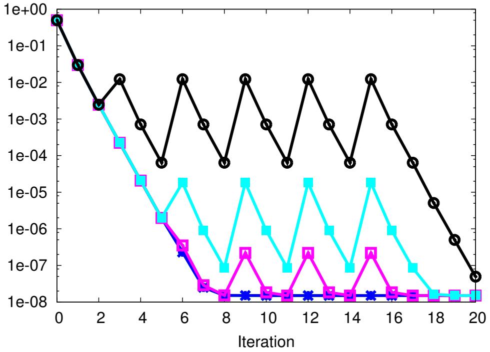
in seconds

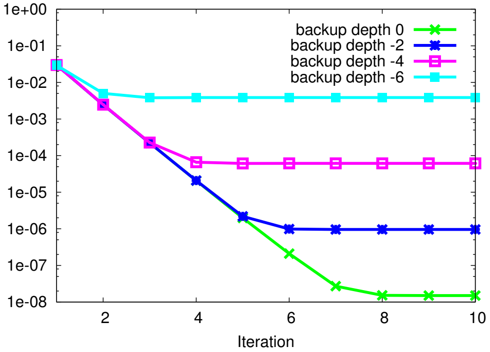
	Zero	MG	SZ_1e-7	SZ_1e-3	aSZ_1e-2	aSZ_1e-1	aSZ_1e0	aSZ_1e1
Fault-iteration: 10								
global	858.41	834.14	807.78	815.00	810.97	813.07	828.05	876.13
local	861.70	815.26	809.89	809.91	810.04	807.84	813.16	828.48
improved	870.76	831.43	830.20	830.15	831.66	830.48	832.72	842.98
iterations	46.33	17.67	18.67	19.00	18.67	18.67	19.00	27.67
Fault-iteration: 40								
global	1061.07	1003.13	815.26	867.33	818.32	816.65	822.13	843.14
local	1022.74	898.64	807.18	824.83	808.21	808.77	806.37	812.24
improved	934.62	850.93	831.36	839.92	831.53	832.23	832.50	834.42
iterations	91.67	29.00	19.67	26.33	19.67	19.67	20.00	22.00
Fault-iteration: 75								
global	1299.50	1240.11	855.73	1092.91	809.08	807.49	823.88	841.65
local	1262.08	1133.81	814.98	973.39	807.41	806.18	806.69	811.86
improved	1005.65	914.84	842.61	909.13	828.58	829.54	829.18	844.56
iterations	141.33	75.00	28.00	72.33	17.33	18.00	18.33	27.33
Fault-iteration: 110								
global	1379.93	1392.10	1066.52	1329.71	810.96	801.84	829.23	837.06
local	1371.13	1352.78	933.86	1197.82	798.91	807.32	806.45	800.54
improved	1062.71	965.28	873.46	944.60	822.93	821.22	824.15	831.35
iterations	190.00	118.33	57.33	106.67	23.00	23.00	24.00	28.33
Average								
global	1149.73	1117.37	886.32	1026.24	812.46	809.76	825.83	849.50
local	1129.41	1050.12	841.48	951.49	806.14	807.52	808.17	813.28
improved	968.44	890.62	844.41	880.95	828.67	828.37	829.64	838.33
iterations	117.33	60.00	30.92	56.08	19.67	19.83	20.33	26.33

Caution: Iterations still indicates an iteration count and not a time measurement.

Multigrid compression

Limits of multigrid compression



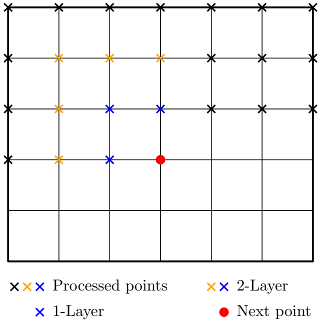


- Discretisation error dominates at some point
- Dominates earlier for highly compressed data
- Factor between L^2-quality and L^2-error depends on amount of repaired data

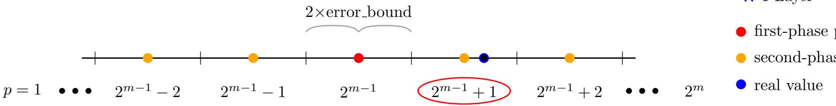
SZ compression (version 1.4.2, 2D)

- Predict values row by row (top to bottom, left to right)
- $\mathcal{V} = \{V(i,j)\}$: set of already compressed point values
- Interpolation based first-phase prediction $f(i,j)$

1-Layer	$V(i,j-1) + V(i-1,j) - V(i-1,j-1)$
2-Layer	$2V(i,j-1)+2V(i-1,j)-4V(i-1,j-1)$ $-V(i,j-2)-V(i-2,j)+2V(i-2,j-1)$ $+2V(i-1,j-2) - V(i-2,j-2)$
$\cdots$	$\cdots$



- 2^m intervals with size of $2 \times \mathtt{eb}_{\mathtt{PW\_REL}}$ around $f(i,j)$



- Store index p or and $p = 0$ and compressed binary-representation if the real value is not in any second-phase prediction interval
- Data is decompressed via interpolation and shifted by the Huffman-code