Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE

Dirk Pflüger
Simulation of Large Systems, IPVS/SimTech, Universität Stuttgart
(joint work with M. Heene, A. Hinojosa)
February 2, 2017
PDE: Turbulence simulations of hot fusion plasmas

- Idea: new, CO$_2$-free source of energy for the generations to come
- EXAHD with H.-J. Bungartz (TUM), M. Griebel (Bonn), T. Dannert (RZG), F. Jenko (UCLA)
Practically Unlimited Resources

Contents:

- Deuterium in bath tub full of water and Lithium in used laptop battery suffice for family over 50 years
Behind the Scenes

- Dilute/hot plasmas are (almost) collisionless
- Not magneto-hydrodynamic, but kinect description (Vlasov):
  \[
  \left[ \frac{\partial}{\partial t} + \vec{v} \frac{\partial}{\partial \vec{x}} + \frac{q}{m} \left( E + \frac{\vec{v}}{c} \times B \right) \frac{\partial}{\partial \vec{v}} \right] f(\vec{x}, \vec{v}, t) = 0
  \]
- Distribution function \( f(\vec{x}, \vec{v}, t) \)
- 6D in state space
- Coupled to Maxwell equations
Behind the Scenes

- Dilute/hot plasmas are (almost) collisionless
- Not magneto-hydrodynamic, but kinect description (Vlasov):
  \[
  \left[ \frac{\partial}{\partial t} + \vec{v} \frac{\partial}{\partial \vec{x}} + \frac{q}{m} \left( E + \frac{\vec{v}}{c} \times B \right) \frac{\partial}{\partial \vec{v}} \right] f(\vec{x}, \vec{v}, t) = 0
  \]
  - Distribution function \( f(\vec{x}, \vec{v}, t) \)
  - 6D in state space
  - Coupled to Maxwell equations
- Gyrokinetics: remove fast gyromotion (smallest scale)
  \[
  \left[ \frac{\partial}{\partial t} + \vec{v} \cdot \frac{\partial}{\partial \vec{x}} + \vec{F} \frac{\partial}{\partial v_{||}} \right] f(\vec{x}, v_{||}, \mu, t) = \Delta(f)
  \]
  - 5D
  - \( \vec{v} \) and \( \vec{F} \) are complex expressions, contain evaluation of \( E \) and \( B \)
Numerical Simulations for Actual Tokamaks with GENE

Aim: global simulations of ITER

State of the art: only small section can be simulated

ASDEX Upgrade

Gyrokinetic Electromagnetic Numerical Experiment

http://www.genecode.org
Numerical Simulations for Actual Tokamaks with GENE

Goal: global simulation with physical realism

- Szenario for simulation of “numerical ITER”
  - Global, non-linear runs
  - At least $10^{11}$ grid points, $10^6$ time steps
  - $>1$ TB just to store single result in memory (complex)

- Possible at all?
Sparse Grids – Hierarchical Approach

- High-dimensional problems suffer “curse of dimensionality”
- Effort $O((2^n)^d) \Rightarrow$ too Big Data

<table>
<thead>
<tr>
<th>full grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d, level 10</td>
</tr>
</tbody>
</table>

Combination technique (multivariate extrapolation-style scheme)

Multiple, but smaller grids:

$O(d \cdot n^d - 1)$ problems of size $O(2^n)$
Sparse Grids – Hierarchical Approach

- High-dimensional problems suffer “curse of dimensionality”
  - Effort $O((2^n)^d) \Rightarrow$ too Big Data
- Therefore: hierarchical discretization
  - Sparse grids: $O(2^n \cdot n^{d-1})$ [Zenger 91]
  - Makes high-dimensional discretizations possible

<table>
<thead>
<tr>
<th>full grid</th>
<th>sparse grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d, level 10</td>
<td>$&gt; 10^{15}$</td>
</tr>
</tbody>
</table>
Sparse Grids – Hierarchical Approach

- High-dimensional problems suffer “curse of dimensionality”
  - Effort $O((2^n)^d)$ $\Rightarrow$ too Big Data
- Therefore: hierarchical discretization
  - Sparse grids: $O(2^n \cdot n^{d-1})$ [Zenger 91]
  - Makes high-dimensional discretizations possible

<table>
<thead>
<tr>
<th>full grid</th>
<th>sparse grid</th>
<th>sg combination technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d, level 10</td>
<td>$&gt; 10^{15}$</td>
<td>25,416,705</td>
</tr>
</tbody>
</table>

- Combination technique (multivariate extrapolation-style scheme)
  - Multiple, but smaller grids: $O(d \cdot n^{d-1})$ problems of size $O(2^n)$
Sparse Grid vs. Combination Technique
Overview

1 Motivation and Numerics

2 Scalability

3 Algorithm-Based Fault Tolerance
   - Hard Faults
   - Silent/Soft Faults

4 Summary
Scalability

Problem of standard solver: global communication within each time-step
Scalability

Problem of standard solver: global communication within each time-step

Use hierarchical ansatz

- Two-level approach
- Numerics: decoupling into locally coupled problems
- Algorithms: second level of parallelism
- First level: no need to scale to exascale
Time-Dependent PDEs

- Gather-scatter steps every time-interval
- Remaining reduced global communication
Global Communication

Optimal communication schemes

Each process group

Distributed full grid

Hierarchize

Distributed hierarchized full grid

Add

Distributed sparse grid

Global reduce

Each component grid

Distributed full grid

Dehierarchize

Distributed hierarchized full grid

Extract

Global communication

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE

Big Data Meets Computation @ IPAM, February 2, 2017
Global Communication

- Minimize number of communications (Range Query Trees):
  \[ O(\log(dn^{d-1})) \times O(2^n n^{d-1}) \]

- Minimize package size
  \[ O(2n \cdot n^{d-1}) \times O(2^{n-1}) \]

- Derivation in BSP/PEM model

[joint work with R. Jacob (ITU, Algorithm Engineering)]
Runtimes on Hazel Hen

**Hierarchization**

- nprocs 1024
- nprocs 2048
- nprocs 4096
- nprocs 8192

**Local Reduction**

- nprocs 1024
- nprocs 2048
- nprocs 4096
- nprocs 8192

**Global Reduction**

- nprocs 1024
- nprocs 2048
- nprocs 4096
- nprocs 8192
Runtimes on Hazel Hen

Total time

![Graph showing runtime vs. total #processes]

- hierarchization + loc. reduction + glob. reduction
- runtime [s]
- nprocs: 1024, 2048, 4096, 8192
- GENE 1 time step

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE

Big Data Meets Computation @ IPAM, February 2, 2017
Overview

1. Motivation and Numerics

2. Scalability

3. Algorithm-Based Fault Tolerance
   - Hard Faults
   - Silent/Soft Faults

4. Summary
Resilience for the Exa-Age

Ever decreasing mean time between failure

- Massive replication of hardware
- Smaller scales (higher integration)
- Hardware possibly with less checks
- ...
Resilience for the Exa-Age

Ever decreasing mean time between failure

- Massive replication of hardware
- Smaller scales (higher integration)
- Hardware possibly with less checks
- ...

Two categories:

1. Hard faults
2. Soft/silent faults
Hard Faults

Errors that trigger signals to the user
- Node, OS, network or process failure
- Software crashes
⇒ Default MPI response: abort application
Hard Faults

Errors that trigger signals to the user

- Node, OS, network or process failure
- Software crashes

⇒ Default MPI response: abort application

Solutions

- Recompute (checkpoint-restart)
  - Checkpoint on HD / RAM
  - Lossless
  - Expensive storage/communication operations
  - Restart even more expensive
Hard Faults

Errors that trigger signals to the user
- Node, OS, network or process failure
- Software crashes
⇒ Default MPI response: abort application

Solutions
- Recompute (checkpoint-restart)
  - Checkpoint on HD / RAM
  - Lossless
  - Expensive storage/communication operations
  - Restart even more expensive
- Continue w/o recomputation
  - Requires adapted numerical schemes
  - No/minor extra computational effort
  - Lossy
⇒ algorithm-based fault-tolerance (ABFT)
Communication Scheme

Master-worker model

Manager Process

Tasks

+ + + - -
+ + + - - +
- -

MPI

MPI

MPI

MPI

Process Group 0

Master

Process Group 1

Master

Process Group 2

Master

Process Group 3

Master
Software Stack

- Fault simulation layer
- Implements interface of ULFM plus `kill_me()` functionality
Selective Reliability

- Focus on critical parts

**Algorithm:** The Combination Technique in Parallel

```plaintext
for all combination grids $\Omega_i$ do in parallel
   $u_i \leftarrow u(x, t = 0)$;  // Set initial conditions
while not converged do
   for all combination grids $\Omega_i$ do in parallel
      $u_i \leftarrow \text{solver}(u_i, N_t)$;  // Solve the PDE on grid $\Omega_i$ ($N_t$ timesteps)
      $u^{(c)}_n \leftarrow \text{reduce}(c_i u_i)$;  // Combine solutions
   for all $i \in I_{n,q,\tau}$ do
      $u_i \leftarrow \text{scatter}(u^{(c)}_n)$;  // Sample each $u_i$ from new $u^{(c)}_n$
```

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE
Big Data Meets Computation @ IPAM, February 2, 2017
Selective Reliability

- Focus on critical parts

**Algorithm:** The Combination Technique in Parallel

```plaintext
for all combination grids $\Omega_i$ do in parallel
  $u_i \leftarrow u(x, t = 0)$ ;                                        // Set initial conditions

while not converged do

  for all combination grids $\Omega_i$ do in parallel
    $u_i \leftarrow \text{solver}(u_i, N_t)$;                           // Solve the PDE on grid $\Omega_i$ ($N_t$ timesteps)
    $u_n^{(c)} \leftarrow \text{reduce}(c_i u_i)$;                   // Combine solutions

  for all $i \in \mathcal{I}_n,q,\tau$ do
    $u_i \leftarrow \text{scatter}(u_n^{(c)})$;                      // Sample each $u_i$ from new $u_n^{(c)}$
```

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE

Big Data Meets Computation @ IPAM, February 2, 2017
Selective Reliability

- Focus on critical parts

**Algorithm:** The Combination Technique in Parallel

```plaintext
for all combination grids $\Omega_i$ do in parallel
  $u_i \leftarrow u(x, t = 0)$;  // Set initial conditions
while not converged do
  for all combination grids $\Omega_i$ do in parallel
    $u_i \leftarrow \text{solver}(u_i, N_t)$;  // Solve the PDE on grid $\Omega_i$ ($N_t$ timesteps)
    mitigateFaults();  // Mitigate faults
    $u_n^{(c)} \leftarrow \text{reduce}(c_i u_i)$;  // Combine solutions
  for all $i \in \mathcal{I}_{n,q,\tau}$ do
    $u_i \leftarrow \text{scatter}(u_n^{(c)})$;  // Sample each $u_i$ from new $u_n^{(c)}$
```

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE

Big Data Meets Computation @ IPAM, February 2, 2017
2D Example
ABFT: Fault-Tolerant Combination Technique

Find alternative combination, exclude missing solutions

- Starting point: standard CT coefficients

\[ u_{\vec{n}}^c(\vec{x}) = \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{\vec{l} \in \mathcal{I}_{\vec{n},q}} u_{\vec{l}}(\vec{x}) \]

In case of failure: use inclusion-exclusion principle to determine adapted combination

1. Solve generalized coefficient problem (GCP):
   \[ \max_{\vec{w} \in \mathbb{R}^d} Q'_{\vec{n}}(\vec{w}), Q'_{\vec{n}}(\vec{w}) := \sum_{\vec{l} \in \mathcal{I}_{\vec{n}}} -\|\vec{i}\|_1 \vec{w}_{\vec{l}}, \text{s.t. } \vec{w}_{\vec{l}} \in \{0, 1\} \forall \vec{l} \in \mathcal{I}_{\vec{n}} \]

2. Obtain new combination coefficients:
   \[ c_{\vec{l}} = (M - 1) \vec{w}_{\vec{l}} \]

Extra computations only on lower scales required
ABFT: Fault-Tolerant Combination Technique

Find alternative combination, exclude missing solutions

- Starting point: standard CT coefficients

\[
u^c_{\vec{n}}(\vec{x}) = \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{\vec{l} \in \mathcal{I}_{\vec{n}, q}} u_{\vec{l}}(\vec{x})
\]

In case of failure: use inclusion-exclusion principle to determine adapted combination
ABFT: Fault-Tolerant Combination Technique

Find alternative combination, exclude missing solutions

- Starting point: standard CT coefficients

\[ u^c_{\vec{n}}(\vec{x}) = \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{\vec{l} \in \mathcal{I}_{\vec{n},q}} u_{\vec{l}}(\vec{x}) \]

In case of failure: use inclusion-exclusion principle to determine adapted combination

1. Solve generalized coefficient problem (GCP):

\[
\max_w Q'(w), \quad Q'(w) := \sum_{l \in \downarrow I} 4^{-\|i\|_1} w_i, \quad \text{s.t. } w_i \in \{0, 1\} \quad \forall l \in \downarrow I
\]
ABFT: Fault-Tolerant Combination Technique

Find alternative combination, exclude missing solutions

- Starting point: standard CT coefficients

\[
u^c_n(\vec{x}) = \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{\vec{l} \in I_n,q} u_l(\vec{x})
\]

In case of failure: use inclusion-exclusion principle to determine adapted combination

1. Solve generalized coefficient problem (GCP):

\[
\max_w Q'(w), \quad Q'(w) := \sum_{l \in I_{\downarrow}} 4^{-\|l\|_1} w_l, \quad \text{s.t. } w_l \in \{0, 1\}, \forall l \in I_{\downarrow}
\]

2. Obtain new combination coefficients:

\[
c_l = (M^{-1} w)_l
\]

- Extra computations only on lower scales required
GCP: 2D Example
GCP: 2D Example
GCP: 2D Example
GCP: 2D Example

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE

Big Data Meets Computation @ IPAM, February 2, 2017
GCP: Example 3D

No faults
GCP: Example 3D

2 faults
GCP: Example 3D

- For arbitrary faults: GCP prohibitively expensive
- Fast solution possible if enough extra grid layers available
- Only fraction of computational effort $\Rightarrow$ faults in lower layers unlikely
- Precompute some extra layers in advance
Results Using GENE

Example:

Good reconstruction (visual inspection)
Results Using GENE (2)

Small (reduced) problem

- 4D: $x, z, \mu, v_\parallel$
- $\vec{l}_\text{min} = [2, 3, 2, 4], \vec{l} = [6, 7, 6, 8] \Rightarrow 69$ combination grids

![Graph showing $L_2$ error norm vs. Faults]

Excellent recovery properties!
Computational Effort

Accumulated time to compute partial grids

Gain by ABFT

Significant savings in runtime
Silent/Soft Faults

No signal to user

- Faults unnoticed unless searched for
- Most common type: Silent Data Corruption (SDC)
  Errors in arithmetic operations, memory corruption, bit flips

\[
\begin{array}{cccccccc}
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
\end{array}
\]
Silent/Soft Faults

No signal to user

- Faults unnoticed unless searched for
- Most common type: Silent Data Corruption (SDC)
  Errors in arithmetic operations, memory corruption, bit flips

Common solutions

- Checksums
- Replication (process/data)
  => Significant overhead (effort, resources)
Selective Reliability

- Focus on critical parts

**Algorithm:** The Combination Technique in Parallel

```plaintext
for all combination grids \( \Omega_i \) do in parallel
  \[ u_i \leftarrow u(x, t = 0); \]  // Set initial conditions

while not converged do
  for all combination grids \( \Omega_i \) do in parallel
    \[ u_i \leftarrow \text{solver}(u_i, N_t); \]  // Solve the PDE on grid \( \Omega_i \) (\( N_t \) timesteps)
    mitigateFaults();  // Mitigate faults
    \[ u_n^{(c)} \leftarrow \text{reduce}(c_i u_i); \]  // Combine solutions
  for all \( i \in I_{n,q,\tau} \) do
    \[ u_i \leftarrow \text{scatter}(u_n^{(c)}); \]  // Sample each \( u_i \) from new \( u_n^{(c)} \)
```

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE
Big Data Meets Computation @ IPAM, February 2, 2017
Selective Reliability

- Focus on critical parts

**Algorithm:** The Combination Technique in Parallel

```plaintext
for all combination grids $\Omega_i$ do in parallel
    $u_i \leftarrow u(x, t = 0)$;  // Set initial conditions
while not converged do
    for all combination grids $\Omega_i$ do in parallel
        $u_i \leftarrow \text{solver}(u_i, N_t)$;  // Solve the PDE on grid $\Omega_i$ ($N_t$ timesteps)
    checkForSDC();  // Cheap sanity check
    mitigateFaults();  // Mitigate faults
    $u_n^{(c)} \leftarrow \text{reduce}(c_i u_i)$;  // Combine solutions
    for all $i \in \mathcal{I}_{n,q,\tau}$ do
        $u_i \leftarrow \text{scatter}(u_n^{(c)})$;  // Sample each $u_i$ from new $u_n^{(c)}$
```

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE
Big Data Meets Computation @ IPAM, February 2, 2017
Silent/Soft Faults

Exploit hierarchical approach

- Similar discretizations lead to similar results
- Exploit redundancy and hierarchical representation to check for faults
- Detection of outliers possible
- Direct integration into communication schemes possible (Subspace Reduce)

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE
Big Data Meets Computation @ IPAM, February 2, 2017
SDC Check: Compare Pairs of Solutions

Similar discretizations should lead to similar results

<table>
<thead>
<tr>
<th>Pair</th>
<th>$\hat{\beta}(s, t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2, 4)</td>
<td>3.98e-01</td>
</tr>
<tr>
<td>(3, 2)</td>
<td>1.11e+00</td>
</tr>
<tr>
<td>(4, 2)</td>
<td>1.11e+00</td>
</tr>
<tr>
<td>(3, 2)</td>
<td>6.32e-01</td>
</tr>
<tr>
<td>(3, 3)</td>
<td>9.85e+05</td>
</tr>
<tr>
<td>(3, 3)</td>
<td>1.07e+06</td>
</tr>
<tr>
<td>(3, 2)</td>
<td>3.98e-01</td>
</tr>
<tr>
<td>(2, 4)</td>
<td>1.27e+00</td>
</tr>
<tr>
<td>(3, 2)</td>
<td>1.07e+06</td>
</tr>
<tr>
<td>(2, 4)</td>
<td>9.85e+05</td>
</tr>
</tbody>
</table>

$$\hat{\beta}(s, t) := \max_{l \leq s \land t} \max_{j \in \mathcal{I}_l} \frac{|\alpha_{l,j}^{(t)} - \alpha_{l,j}^{(s)}|}{\min \{ |\alpha_{l,j}^{(t)}|, |\alpha_{l,j}^{(s)}| \}}$$
SDC Check: Outlier detection

\[ \tilde{u}(0, 0) = [1.002, 5.356, 0.998, 1.002, 1.001, 1.001, .999] \]
2D Example

Advection equation

\[
\frac{\partial u}{\partial t} + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} = 0 \quad \Omega = [0, 1]^2
\]

- Periodic boundary conditions
- Constant advection velocities \(c_x, c_y\)
- Initial condition \(u(x, y, t = 0) = \sin(2\pi x) \sin(2\pi y)\)
- Lax-Wendroff scheme (2nd order space + time)
- Error/solution at \(t = 0.5\) compared to analytical solution
  \[
  u(x, y, t) = \sin(2\pi (x - c_x t)) \sin(2\pi (y - c_y t))
  \]
- Corruption of one single data point in initial condition
2D Example

- **Exact solution**
- **Full Grid**
- **Combined grid**

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE

Big Data Meets Computation @ IPAM, February 2, 2017
2D Example

Exact solution

Full Grid

Combined grid

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE

Big Data Meets Computation @ IPAM, February 2, 2017
2D Example
2D Example: Simulated Soft Faults

- Inserting one soft fault
- Measuring L2-error at the end

\[ \tilde{u}_i(x_{l1,j1}, x_{l2,j2}) = u_i(x_{l1,j1}, x_{l2,j2}) \times 10^{-300} \]

\[ \tilde{u}_i(x_{l1,j1}, x_{l2,j2}) = u_i(x_{l1,j1}, x_{l2,j2}) \times 10^5 \]
Higher-D: Advection-Diffusion Equation

\[ \partial_t u - \Delta u + \bar{a} \cdot \nabla u = f \quad \text{in } \Omega \times [0, T) \]
\[ u(\cdot, t) = 0 \quad \text{in } \partial \Omega \]
\[ u(\cdot, 0) = u_0 \quad \text{in } \Omega \]

\[ \Omega = [0, 1]^d, \bar{a} = (1, \ldots, 1)^T, u_0 = e^{-100 \sum_{i=1}^{d} (x_i - 0.5)^2} \]

- Implemented in DUNE-pdelab
- FVM, explicit time integration
Results

- Fault in second time step
- Relative error w.r.t. full-grid solution ($n = 11$ in 2D, $n = 7$ in 5D)
- Computations on Hazel Hen (HLRS)
- 2D, 5D:

```
(6,6)  (7,7)  (8,8)  (9,9)  (10,10)

$10^{-2}$  $10^{-1}$

$l_2$-error

Δ no faults
○ 2 groups (~50% faults)
■ 4 groups (~25% faults)
★ 8 groups (~12% faults)
× 16 groups (~6% faults)

(4,4,4,4)  (5,5,5,5)  (6,6,6,6)

$10^{0}$

$l_2$-error

Δ no faults
○ 2 groups (~50% faults)
■ 4 groups (~25% faults)

Again: excellent recovery properties!

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE
Big Data Meets Computation @ IPAM, February 2, 2017
Overview

1 Motivation and Numerics

2 Scalability

3 Algorithm-Based Fault Tolerance
   - Hard Faults
   - Silent/Soft Faults

4 Summary
Summary

Gyrokinetics

- High-dimensional problem with urgent need for compute resources
Summary

Gyrokinetics

- High-dimensional problem with urgent need for compute resources

Hierarchical multilevel splitting provides novel handles on exa-challenges

- **Scalability**
  - 2nd level of parallelism
  - Numerical decoupling, extrapolation
  - Exploit hierarchical splitting for optimal communication

- **ABFT at low cost**
  - Exploit hierarchical scheme
  - Recombination rather than recomputation

- **Silent faults**
  - Exploit underlying hierarchical basis
  - Detection and treatment of silent faults possible
Summary

Gyrokinetics

- High-dimensional problem with urgent need for compute resources
- Sparse grids: "Too Big Data" ⇒ Big Data

Hierarchical multilevel splitting provides novel handles on exa-challenges

- **Scalability**  
  - 2nd level of parallelism  
  - Numerical decoupling, extrapolation  
  - Exploit hierarchical splitting for optimal communication

- **ABFT at low cost**  
  - Exploit hierarchical scheme  
  - Recombination rather than recomputation

- **Silent faults**  
  - Exploit underlying hierarchical basis  
  - Detection and treatment of silent faults possible

Dirk Pflüger: Scalability and Algorithm-Based Fault Tolerance for Plasma Physics Simulations with GENE
Thanks to:

... and all others!
Thanks to:

... and all others!

Thank you for your interest!
Mario Heene, Alfredo Parra Hinojosa, Hans-Joachim Bungartz, and Dirk Pfüger.
A massively-parallel, fault-tolerant solver for time-dependent pdes in high dimensions.
Accepted.

Philipp Hupp, Mario Heene, Riko Jacob, and Dirk Pfüger.
Global communication schemes for the numerical solution of high-dimensional PDEs.

Mario Heene and Dirk Pfüger.
Scalable algorithms for the solution of higher-dimensional PDEs.

Alfredo Parra Hinojosa, Christoph Kowitz, Mario Heene, Dirk Pfüger, and Hans-Joachim Bungartz.
Towards a fault-tolerant, scalable implementation of GENE.

Alfredo Parra Hinojosa, Brendan Harding, Hegland Markus, and Hans-Joachim Bungartz.
Handling silent data corruption with the sparse grid combination technique.

Dirk Pfüger, Hans-Joachim Bungartz, Michael Griebel, Frank Jenko, Tilman Dannert, Mario Heene, Alfredo Parra Hinojosa, Christoph Kowitz, and Peter Zaspel.
EXAHD: An exa-scalable two-level sparse grid approach for higher-dimensional problems in plasma physics and beyond.