EXAHD – Current Work on Scalable and Fault-Tolerant Plasma Simulations

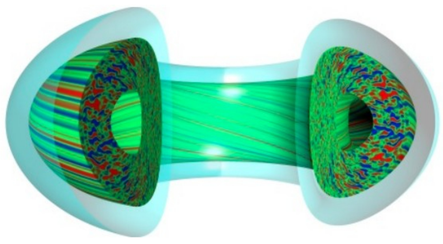
University of Stuttgart
Institute for Parallel and Distributed Systems (IPVS)

theresa.pollinger@ipvs.uni-stuttgart.de

Theresa Pollinger, Dirk Pflüger, Michael Obersteiner, Hans-Joachim Bungartz, Johannes Rentrop, Michael Griebel, Rafael Lago, Tilman Dannert, Frank Jenko

Approach

Solve a higher-dimensional 5(+1)**-d** gyrokinetic system for the simulation of fusion plasma microturbulence



$$\frac{\partial F}{\partial t} + \dot{\mathbf{X}} \cdot \nabla F + \dot{v}_\parallel \frac{\partial F}{\partial v_\parallel} + \dot{\mu} \frac{\partial F}{\partial \mu} = 0$$

w.r.t. the distribution function $F(x, y, z, v_\parallel, \mu, t)$

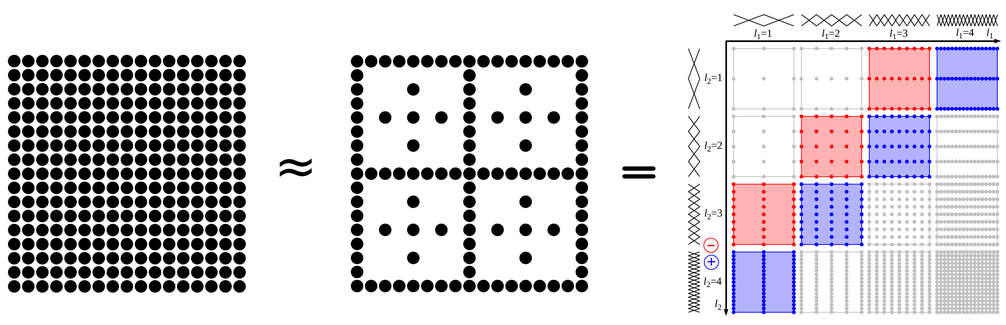
Our aim: to globally solve the gyrokinetic equations for ITER

But: Standard simulations in higher d suffer the **curse of dimensionality**

$N = \frac{1}{h}$, $\#points \sim \mathcal{O}(N^d)$, $error \sim \mathcal{O}(N^{-2})$ (typically)

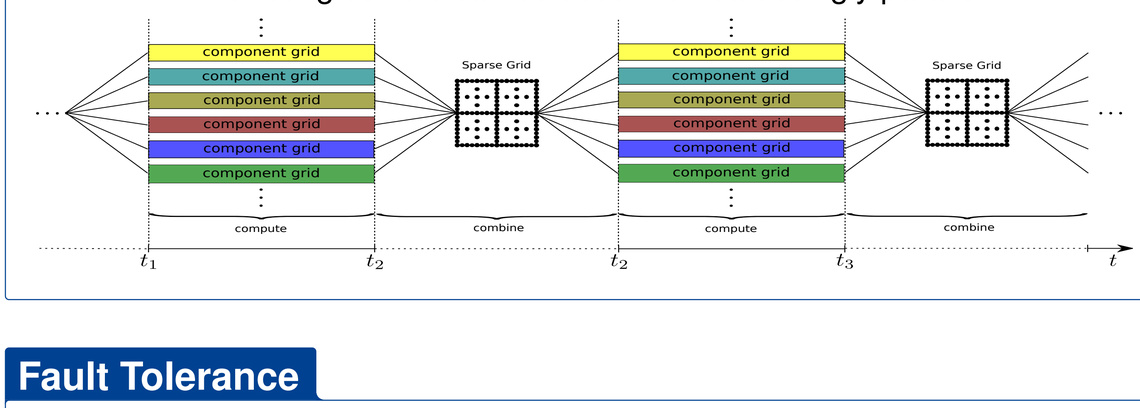
Sparse Grid Combination Technique to numerically decouple problem: combine partial solutions on regular grids

$\#points \sim \mathcal{O}(d(\log N)^{d-1}) \cdot \mathcal{O}(N)$, $error \sim \mathcal{O}(N^{-2}(\log N)^{d-1})$



$$u(x) \approx \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{|l|=n-q} u_l(x)$$

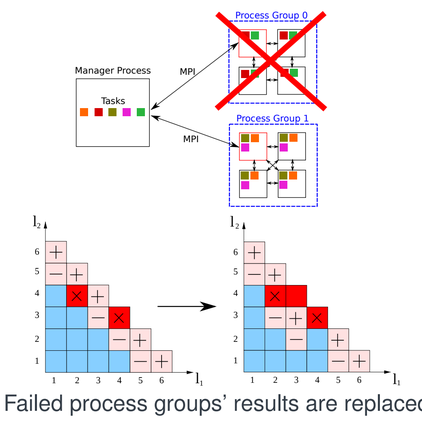
The partial solutions $u_l(x)$ are **independent**
$\Rightarrow$ existing solvers can be used – embarassingly parallel



Fault Tolerance

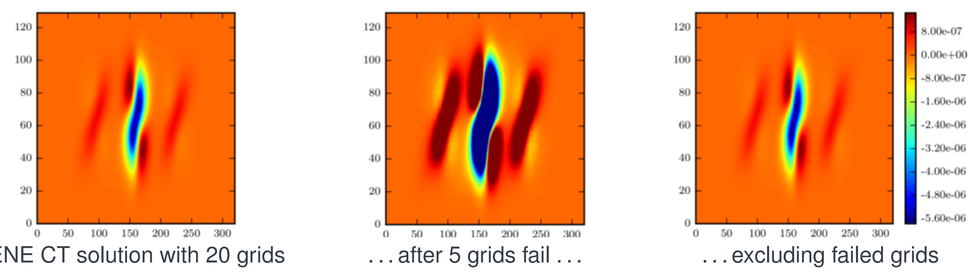
Hardware failures are becoming a problem for exascale.

Fault-Tolerant Combination Technique:
algorithm-based fault tolerance for hard and soft faults – no checkpoint/restart



Failed process groups' results are replaced

while *not converged*
  for all *CT grids* Ω_l **do in parallel**
    $u_l = \text{solver}(u_l, N_t)$; `// solve` N_t `steps`
    decideToKill() ; `// hard faults`
    decideToChange() ; `// soft faults`
  checkForSDC(); `// sanity check`
  recover(); `// mitigate faults`
  $u_n^{(c)} = \text{reduce}(c_l u_l)$; `// combine`
  forall $l \in \mathcal{I}_{n,q}$ **do**
    $u_l = \text{scatter}(u_n^{(c)})$; `// re-distribute`



GENE CT solution with 20 grids | . . . after 5 grids fail . . . | . . . excluding failed grids

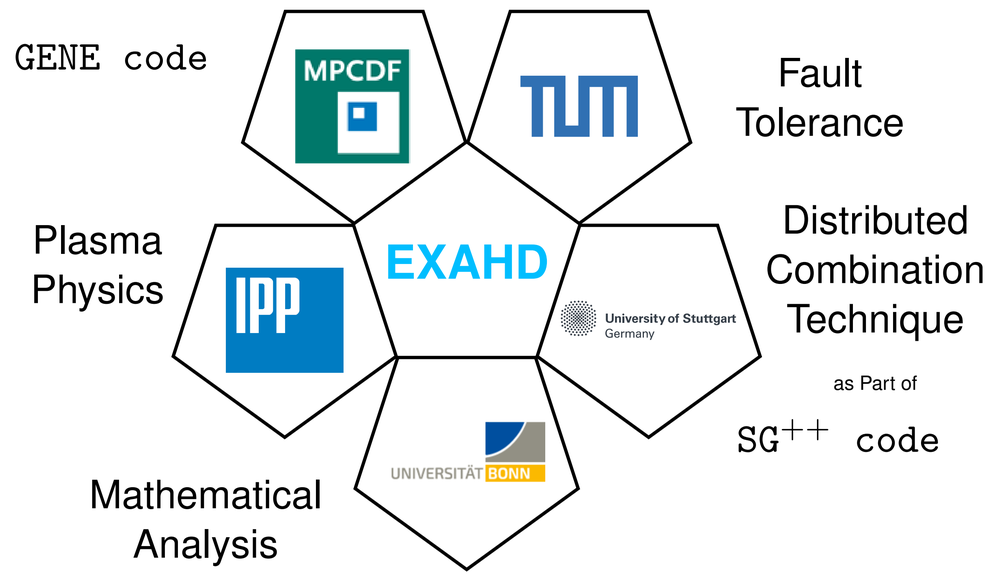
$\Rightarrow$ **good approximation** even after recovery

The Project

Vision

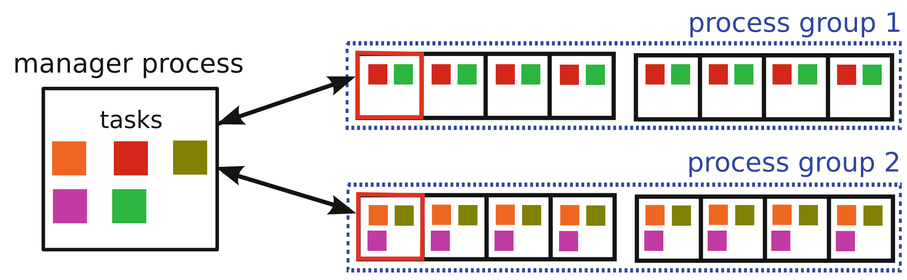
- Algorithms fit to Exascale: Fault tolerance for higher-dimensional problems on all levels of parallelization, and scaling beyond single HPC systems.
- Software Framework: A general tool for the solution of higher-dimensional problems, with efficient adaptive and dynamic load balancing.
- Numerical ITER: Drive GENE code to scenarios that are far beyond what is currently feasible.

Team



Scaling with an Extra Level of Parallelism

In addition to GENE MPI & OpenMP: Manager-Worker Pattern



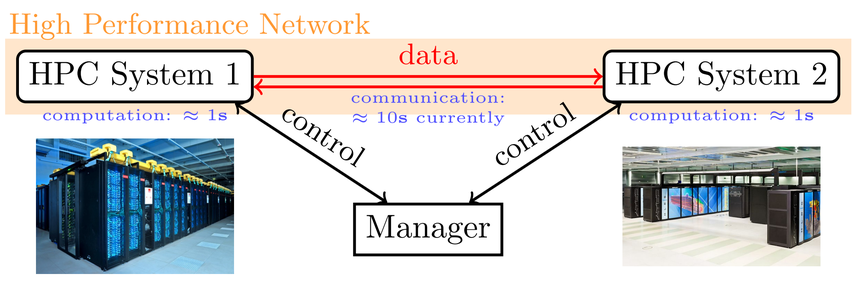
Parallelization across partial solutions

Performance

The combination technique overhead scales at a level negligible to the GENE set-up

WIP: Scaling Beyond Single HPC Systems

As the problem sizes get larger, why not **couple more HPC Systems?**



Publications and More

At `ipvs.informatik.uni-stuttgart.de/SGS/EXAHD/`

SPPEXA

2013 SPPEXA phase 1 2016 SPPEXA phase 2 2020

Linear, local scenarios | Massive parallelism, load balancing | Algorithm-based fault tolerance | Non-linear global scenarios | Beyond MPI | Numerical ITER

DFG Deutsche Forschungsgemeinschaft
SPP 1648