HPC and HPDA Projects in DLR Aeronautics and Space Research

Dr.-Ing. Achim Basermann

German Aerospace Center (DLR), Cologne
Simulation and Software Technology
Head of Department „High-Performance Computing“
Survey

• Our Department SC-HPC at DLR

• Exascale Computing and Performance Engineering

• Big Data & High Performance Machine Learning
  • Space Debris Management
  • Rocket Engine Combustion Analysis

• Multi-Disciplinary Optimization

• Quantencomputing
German Aerospace Center

• Research Institution
  • Research areas: aeronautics; space research and technology; transport; energy; digitalization; defence and security
  • national and international cooperations

• Space Agency

• Project Management Agency
Approx. 8600 employees across 47 institutes and facilities at 27 sites.

DLR Simulation and Software Technology

- stands for **innovative software engineering**, 
- develops **challenging individual software solutions** for DLR, and 
- is partner in **scientific projects** in the area of simulation and software technology.
High Performance Computing Teams

Department
High Performance Computing
Head: Dr. Achim Basermann
Deputy: Dr. Margrit Klitz

Intelligent Algorithms & Optimization
Dr. Martin Siggel

Parallel Numerics
Dr. Jonas Thies

Quantum Computing
Dr. Tobias Stollenwerk
Exascale computing and Performance Engineering
ESSEX goes Oakforest-PACS

Graphvisualisierung der möglichen Zustandsänderungen des Heisenberg Spinkettenmodells
Motivation: Requirements for Exascale Computing

Quantum physics/information applications

Large, Sparse


\[ i\hbar \frac{\partial}{\partial t} \psi(\vec{r},t) = H\psi(\vec{r},t) \]

and beyond....

\[ H \mathbf{x} = \lambda \mathbf{x} \]

“Few” (1,…,100s) of eigenpairs

“Bulk” (100s,…,1000s) eigenpairs

\[ \{\lambda_1, \lambda_2, \ldots, \ldots, \lambda_k, \ldots, \ldots, \ldots, \lambda_{n-1}, \lambda_n\} \]

Good approximation to full spectrum (e.g. Density of States)

→ Sparse eigenvalue solvers of broad applicability
The ESSEX Software Infrastructure: MPI + X with

- System with multiple CPUs (NUMA domains) and GPUs
The ESSEX Software Infrastructure: MPI + X with

- System with multiple CPUs (NUMA domains) and GPUs
- -np 1: use entire CPU
The ESSEX Software Infrastructure: MPI + X with

- System with multiple CPUs (NUMA domains) and GPUs
  - `-np 1`: use entire CPU
  - `-np 2`: use CPU and first GPU
The ESSEX Software Infrastructure: MPI + X with

- System with multiple CPUs (NUMA domains) and GPUs
- -np 1: use entire CPU
- -np 2: use CPU and first GPU
- -np 3: use CPU and both GPUs
The ESSEX Software Infrastructure: MPI + X with

- System with multiple CPUs (NUMA domains) and GPUs
  -np 1: use entire CPU
  -np 2: use CPU and first GPU
  -np 3: use CPU and both GPUs
  -np 4: use one process per socket and one for each GPU

Option: distribute problem according to memory bandwidth measured
Application, Algorithm and Performance: Kernel Polynomial Method (KPM) – A Holistic View

- Compute **approximation to the complete eigenvalue spectrum** of large sparse matrix $A$ (with $X = I$)

\[
X(\omega) = \frac{1}{N} \text{tr}[\delta(\omega - H)X] = \frac{1}{N} \sum_{n=1}^{N} \delta(\omega - E_n) \langle \psi_n, X \psi_n \rangle
\]
The Kernel Polynomial Method (KPM)

Optimal performance exploit knowledge from all software layers!

Basic algorithm – Compute Cheyshev polynomials/moments:

```plaintext
for r = 0 to R - 1 do
  |v⟩ ← |rand(0)

Initialization steps and computation of η₀, η₁
for m = 1 to M/2 do
  swap(|w⟩, |v⟩)
  |u⟩ ← H|v⟩
  |u⟩ ← |u⟩ - b|v⟩
  |w⟩ ← −|w⟩
  |w⟩ ← |w⟩ + 2a|u⟩
  η₂m ← ⟨v|v⟩
  η₂m+1 ← ⟨w|v⟩
end for
end for
```

Application:
Loop over random initial states

Algorithm:
Loop over moments

Building blocks:
(Sparse) linear algebra library

Sparse matrix vector multiply
Scaled vector addition
Vector scale
Scaled vector addition
Vector norm
Dot Product
The Kernel Polynomial Method (KPM)

Optimal performance exploit knowledge from all software layers!

Basic algorithm – Compute Cheyshev polynomials/moments:

\[
\begin{align*}
\text{for } & \ r = 0 \text{ to } R - 1 \text{ do} \\
& |v\rangle \leftarrow |\text{rand}(0)\rangle \\
& \text{Initialization steps and computation of } \eta_0, \eta_1 \\
\text{for } & \ m = 1 \text{ to } M/2 \text{ do} \\
& \text{swap}(|w\rangle, |v\rangle) \\
& |u\rangle \leftarrow H|v\rangle \\
& |u\rangle \leftarrow |u\rangle - b |v\rangle \\
& |w\rangle \leftarrow -|w\rangle \\
& |w\rangle \leftarrow |w\rangle + 2a |u\rangle \\
& \eta_{2m} \leftarrow \langle v|v\rangle \\
& \eta_{2m+1} \leftarrow \langle w|v\rangle
\end{align*}
\]

Augmented Sparse Matrix Vector Multiply
The Kernel Polynomial Method (KPM)

Optimal performance exploit knowledge from all software layers!

Basic algorithm – Compute Chebyshev polynomials/moments:

\[
\begin{align*}
\text{for } r = 0 \text{ to } R - 1 \text{ do} \\
|v^r\rangle \leftarrow |\text{rand}\rangle \\
\text{Initialization steps and computation of } \eta_0, \eta_1 \\
\text{for } m = 1 \text{ to } M/2 \text{ do} \\
\text{swap}(|w\rangle, |v\rangle) \\
|w\rangle = 2a(H - b\mathbb{1})|v\rangle - |w\rangle & \& \\
\eta_{2m} = \langle v|v\rangle & \& \\
\eta_{2m+1} = \langle w|v\rangle & \text{aug_spmmv()}
\end{align*}
\]

\[
\begin{align*}
|V\rangle := |v\rangle_0..R-1 \\
|W\rangle := |w\rangle_0..R-1 \\
|V\rangle \leftarrow |\text{rand}\rangle \\
\text{Initialization steps and computation of } \mu_0, \mu_1 \\
\text{for } m = 1 \text{ to } M/2 \text{ do} \\
\text{swap}(|W\rangle, |V\rangle) \\
|W\rangle = 2a(H - b\mathbb{1})|V\rangle - |W\rangle & \& \\
\eta_{2m}[:i] = \langle V|V\rangle & \& \\
\eta_{2m+1}[:i] = \langle W|V\rangle & \text{aug_spmmv()}
\end{align*}
\]

Augmented Sparse Matrix
Multiple Vector Multiply
KPM: Heterogenous Node Performance

- Topological Insulator Application
- Double complex computations
- Data parallel static workload distribution
KPM: Large Scale Heterogenous Node Performance

Performance Engineering of the Kernel Polynomial Method on Large-Scale CPU-GPU Systems

*Thanks to CSCS/T. Schulthess for granting access and compute time
**Scalability on Oakforest-PACS**

*seit 6 / 2018 auf Platz Nummer 12 der TOP 500*

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores</td>
<td>556,104</td>
</tr>
<tr>
<td>Memory</td>
<td>919,296 GB</td>
</tr>
<tr>
<td>Processor</td>
<td>Intel Xeon Phi 7250 68C 1.4GHz (KNL)</td>
</tr>
<tr>
<td>Interconnect</td>
<td>Intel Omni-Path</td>
</tr>
<tr>
<td>Linpack Performance (Rmax)</td>
<td>13.554 PFlop/s</td>
</tr>
<tr>
<td>Theoretical Peak (Rpeak)</td>
<td>24.913 PFlop/s</td>
</tr>
<tr>
<td>HPCG TFlop/s</td>
<td>9,938,880 385.479</td>
</tr>
</tbody>
</table>

Impression of the Oakforest-PACS supercomputer at the Japanese joint center for advanced HPC (JCAHPC).
Large scale performance – weak scaling

Computing 100 inner eigenvalues on matrices up to $n = 4 \times 10^9$

Typical Application[1]: Topological Insulator

Large scale performance – weak scaling

Computing 100 inner eigenvalues on matrices up to $n = 4 \times 10^9$

SUPERMUC (SNG)
Leibniz Supercomputing Centre (LRZ) in Garching
6480 CPU-only dual-socket nodes with Intel Skylake-SP
311,040 compute cores

Weak scaling of BEAST-P on SNG for problems of size $2^{21}$ (1 node) to $1.53 \times 2^{32}$ (3136 nodes, about half of the full machine)
How to ensure the quality of the ESSEX software: Basics

- **Git** for distributed software development
  - **Merge-request workflow** for code review; changes only in branches
  - Visualization of git repository development
- Own MPI extension for **Google Test**
- Realization of **continuous-integration** with Jenkins server
Towards common standards and community software for extreme-scale computing

As we approach the Exa-scale, requirements on robustness, portability, scalability and interoperability of scientific software are rapidly increasing

**xSDK: Extreme-scale Scientific Software Development Kit**

- Joint open-source effort of DOE labs and other international teams ([https://xsdk.info/](https://xsdk.info/))

- DLR contributes a hybrid-parallel library for solving sparse eigenvalue problems on heterogenous supercomputers
  - ([https://bitbucket.org/essex/phist/](https://bitbucket.org/essex/phist/))
Towards common standards and community software for extreme-scale computing
Big Data & High Performance Machine Learning
Big Data @ DLR

How to perform data analytics on huge datasets?
Example 1: Space Debris Management

Picture from http://kidsnews.hu/2018/03/az-urszemetrol/
The space debris problem
Space Debris

All non-active, non-cooperating orbital objects like
• Old or defect satellites
• Lost Tools
• Debris of all kind (e.g. from satellite collisions)

• Impose danger already from 1 cm size
• ~1 000 000 objects, around 18.000 tracked in database

Simulated collision of projectile with 7 km/s on aluminum plate. Picture: ESA.
Our solution: Software BACARDI

- Database of all orbits from known space objects (Group „Space Situational Awareness”)
- Methods being used:
  - Orbit determination of 1,000,000 objects
  - Propagation
  - Object identification
  - Collision prediction for mission support
- “Bacardi Viewer“ allows a 3D visualization of objects from the BACARDI database

Tracking of space debris with lasers
Image source: https://www.wired.com/2011/10/space-junk-laser/
Example 2:
Rocket engine combustion analysis

**Goal:** Cost reduction of rocket engines, be competitive with e.g. Space-X

### Hybrid rocket engine

- Pressurized fluid oxidizer
- Solid fuel
- A valve controls, how much oxidizer gets into the combustion chamber

**Advantages**
- Cheap
- Controllable
Example 2:
Rocket engine combustion analysis

- **Goal**: Cost reduction of rocket engines, be competitive with e.g. Space-X

**Hybrid rocket engine**

- Pressurized fluid oxidizer
- Solid fuel
- A valve controls, how much oxidizer gets into the combustion chamber

- Advantages
  - Cheap
  - Controllable

**Question**: Can we detect problems / inefficiencies during combustion?

**Challenge**: High speed camera produces huge data sets
Our solution: Software HeAT!

- **HeAT = Helmholtz Analytics Toolkit**
- Python framework for **parallel, distributed** data analytics and machine learning
- Developed within the Helmholtz Analytics Framework Project since 2018
- AIM: Bridge data analytics and **high-performance computing**
- Open Source licensed, MIT

[helmholtz-analytics/heat](https://github.com/helmholtz-analytics/heat)
How we started HeAT: The Helmholtz Analytics Framework (HAF) Project

• Joint project of all 6 Helmholtz centers

• Goal: foster data analytics methods and tools within Helmholtz federation.

• Scope:
  • Development of domain-specific data analysis techniques
  • Co-design between domain scientists and information experts
Motivation: HAF applications

Earth System Modelling

SEVIRI Satellite Images – Near Real Time

Research with Photons

Neuroscience

Aeronautics and Aerodynamics

Structural Biology
Greatest Common Denominator?

Machine Learning
= Data
+ Numerical Linear Algebra

https://xkcd.com/1838/
**Scope**

- Facilitating applications of HAF in their work
- Bringing HPC and Machine Learning / Data Analytics closer together
- Ease of use

**Design**

- HeAT
  - k-means
  - SVM
  - Deep Learning
  - And more machine learning algorithms
- PyTorch
  - Tensor Linear Algebra
  - Automatic Differentiation
  - NumPy-like interface
  - GPU support
- mpi4py
  - Distributed Parallelism (MPI)
Data analysis with K-means clustering

Idea: Separate dataset into k distinct clusters

1. Start with random cluster centroids (crosses)

2. Compute distance of every point to all centroids

3. Assign point to closest centroid

4. Update centroid positions at cluster center
Data analysis with K-means clustering

Numpy vs. HeAT

NumPy

```python
>>> for i in range(self.n_clusters):
    new_centroids[:, :, i:i+1] = ((data*selection).sum(axis=0, keepdims=True) / 
                                selection.sum(axis=0).clip(1.0, sys.maxsize))
```

HeAT

```python
>>> for i in range(self.n_clusters):
    new_centroids[:, :, i:i+1] = ((data*selection).sum(axis=0) / 
                                selection.sum(axis=0).clip(1.0, sys.maxsize))
```

HeAT hides parallelism, looks like sequential NumPy code.
Combustion clustering results:
Resulting Clusters, $k = 7$
Computational Performance

• Hybrid shared memory + distributed memory setting

• Variation of 1 … 16 MPI total ranks (processing units)

• How does the computing time reduce with number of processing units?

• First results look promising, testing on larger systems + graphic cards necessary
SC-HPC Highlights in the DLR Project ATEK: Propulsion Technologies and Components for Carrier Systems

Contribution of HPC:

- Data analysis (e.g. clustering) of 300k images from rocket engine combustion experiments.
- Results validated with HeAT (Helmholtz Analytics Toolkit).
- Further work: - optimization of emission spectrum simulations
  - data fusion in thermo-mechanical experiments

- ATEK research rocket was launched on June 13th 2019.
- The rocket reached an altitude of 239 km.
- It landed in a distance of 67 km from Esrange Space Center.
- All experiments were successful.
HeAT software: Transparent development process

Github for code review, issue tracking, sprint planning

Travis for continuous integration

Mattermost for discussions

https://github.com/helmholtz-analytics

Join us there!
Multi-Disciplinary Optimization
Analysis and Optimization of the Spaceliner Pre-Design

- Development of a hypersonic passenger spacecraft for long distance flights

- Descent should be accomplished in gliding flight

- **New research focus:** development of a hybrid structure with integrated thermal control units involving magnetohydrodynamic (MHD) effects with cooled magnets
Implementation of a Multidisciplinary Optimization Loop

- Implementation of the design as process graph in the software platform RCE (remote component environment) by coupling tools from different disciplines

- Problem: no derivatives available

- Up to now: use of derivative-free optimizers from toolbox DAKOTA

- **Our development**: new algorithm for nonlinear derivative-free constrained optimization
  - Derivative-free trust-region SQP-method
Quantencomputing
Quantum Computers for Solving Aerospace Problems

Challenges:
• Quantum computer interfaces are close to hardware
• Which quantum algorithms for which applications are superior to classical computing?

DLR QC Research:
• Investigate algorithms and applications for near-term QC devices
• Develop tools and algorithms to use QC-devices
• Perform experiments on early QC devices

Google QC Chip
D-Wave Q. Annealer

Programming Quantum Computers
(arxiv:1612.08091)
Adiabatic Quantum Computer

- Optimizer for quadratic unconstrained binary problems (QUBO)

\[ E = \sum_i H_i \cdot x_i \quad \text{with } x_i \in \{0, 1\} \]
Adiabatic Quantum Computer

- Optimizer for quadratic unconstrained binary problems (QUBO)

\[ E = \sum_i H_i \ x_i \]

with \( x_i \in \{0, 1\} \)
Adiabatic Quantum Computer

- Optimizer for quadratic unconstrained binary problems (QUBO)

\[ E = \sum_i H_i x_i \]  

with \( x_i \in \{0, 1\} \)
Adiabatic Quantum Computer

- Optimizer for quadratic unconstrained binary problems (QUBO)

\[ E = \sum_i H_i \ x_i + \sum_{i \neq j} J_{ij} \ x_i x_j \quad \text{with} \quad x_i \in \{0, 1\} \]
Adiabatic Quantum Computer

- Optimizer for quadratic unconstrained binary problems (QUBO)

\[ E = \sum_i H_i x_i + \sum_{i \neq j} J_{ij} x_i x_j \quad \text{with } x_i \in \{0, 1\} \]
Adiabatic Quantum Computer

Example:

\[ E = 5x_1 + 2x_2 - 3x_3 - x_1x_2 + 3x_2x_3 - 4x_3x_1 \]

Lowest Energy: \( E = -3 \)

at \((x_1, x_2, x_3) = (0, 0, 1)\)

- Quantum systems have discrete energy levels (e.g. atom)

- Idea: Find system whose lowest energy state (ground state) corresponds to the solution of the optimization problem
Adiabatic Quantum Computer

Example:

\[ E = 5x_1 + 2x_2 - 3x_3 - x_1x_2 + 3x_2x_3 - 4x_3x_1 \]

Lowest Energy: \( E = -3 \)

at \( (x_1, x_2, x_3) = (0, 0, 1) \)

- Quantum systems have discrete energy levels (e.g. atom)
- Idea: Find system whose lowest energy state (ground state) corresponds to the solution of the optimization problem
Applications for Quantum Annealers

Applications

Which problems can be mapped to QUBO?

\[ E = \sum_i H_i x_i + \sum_{i \neq j} J_{ij} x_i x_j \quad \text{with } x_i \in \{0, 1\} \]

- All NP-Complete Problems. E.g.
  - Graph Partitioning
  - Satisfiability Problems
- Planning
  - Job-Shop Scheduling
  - Mars-Lander Operations
- Machine Learning

* Venturelli et. al. arXiv:1506.08479
** Rieffel et. al. arXiv:1407.2887
Quantum Computing for Flight Gate Assignment

- Optimal assignment of flights to gates at airports
- Hard combinatorial optimization problem
- Amenable to D-Wave quantum annealer and gate based quantum computers (Google, IBM, etc.)
- Use real world data and perform experiments on hardware as well as simulations
- Collaboration with NASA Ames

Optimization with gate-based QC (QAOA)
Quantum Computing for Radar Cross Section Calculation

- Algorithm for solving linear systems of equations (HHL)
- Amenable to large gate-based quantum computers
- Exponential speed-up over classical computers (if certain conditions are fulfilled)
- Goal: Resource estimation

FEM calculation for radar cross section
(Clader et.al. arXiv:1301.2340)

HHL Algorithm
(Clader et.al. arXiv:1301.2340)
Quantum Computing for Earth Observation Data Acquisition Planning

- Optimal planning of earth observation imaging acquisition
- Hard combinatorial optimization problem
- Only solvable with heuristics
- Minor improvements to solutions would have strong impact
- Amenable to D-Wave Quantum Annealer and gate based quantum computers
- Collaboration with Airbus

Earth Observation Satellite
Deconflicting Flights with Quantum Annealers

- Wind-optimal flight trajectories show conflicts
- Resolve these conflicts
- Reduce flight delays
- Amenable to D-Wave quantum annealer
- Collaboration with NASA Ames

Conflicts of transatlantic flights

Time-to-Solution on D-Wave
(arXiv:1711.04889)
Many thanks for your attention!

Questions?

Dr.-Ing. Achim Basermann
German Aerospace Center (DLR)
Simulation and Software Technology
Department Head High-Performance Computing
Achim.Basermann@dlr.de
http://www.DLR.de/sc

We are hiring: http://www.dlr.de/jobs/