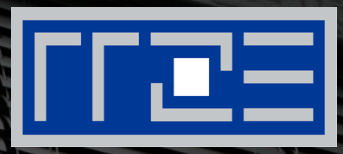
REGIONALES RECHENZENTRUM ERLANGEN [RRZE]



High Performance Computing

Systemausbildung − Grundlagen und Aspekte von Betriebssystemen und System-nahen Diensten

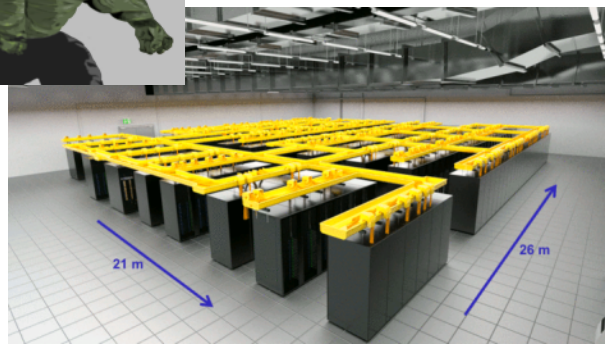
Michael Meier, RRZE, 01.07.2015

Agenda

- Was bedeutet HPC?
- Aus welchen Komponenten besteht ein typisches HPC-System?
- Mit dem HPC-System läuft $GTA5/$Photoshop/ $Videoschnittprogramm doch superschnell, oder?
- Shared-Memory-Parallelisierung vs. Message Passing
- Was ist dieses Queuing-System und wozu ist es gut?

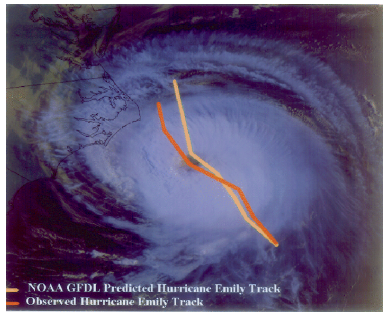
Was bedeutet HPC?

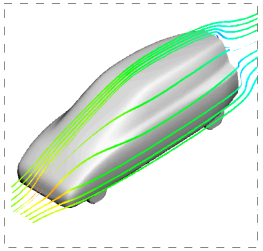
- Englisch: High Performance Computing
- Deutsch: **Hochleistungsrechnen**
- “Hochleistungsrechnen ist ein Bereich des computergestützten Rechnens. Er umfasst alle Rechenarbeiten, deren Bearbeitung einer hohen Rechenleistung oder Speicherkapazität bedarf.“ (Wikipedia)
- genaue Abgrenzung ist schwierig

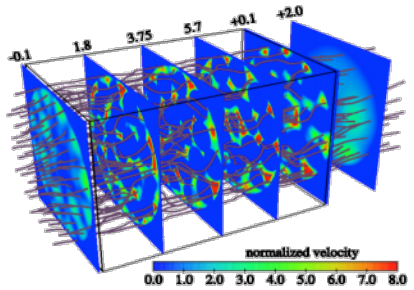


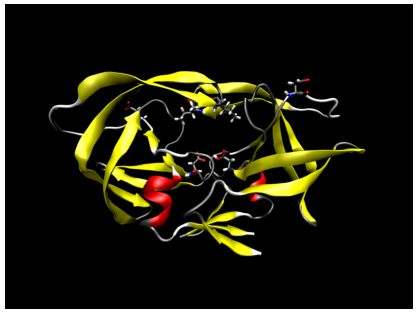
Einsatzbereiche von HPC-Systemen (Beispiele)

- Ingenieurwesen (Automotive)
  - Crashsimulation
  - Aerodynamik
  - Akustik
- Meteorologie
  - Wettervorhersage
  - Katastrophenvorhersage
- Biologie/Medizin
  - “Drug Design”
  - Aufklärung von Reaktionsprozessen









Einsatzbereiche von HPC-Systemen (Beispiele)

- Werkstoffwissenschaften
- Physik
  - Aufklärung von fundamentalen Wechselwirkungen
  - Struktur der Materie
- Filmindustrie
  - Rendering
  - CGI

- Zunehmend auch außerhalb der klassischen Natur-/Ingenieurswissenschaften, z.B.:
  - Wirtschaft
  - Sprachwissenschaften

HPC-Systeme

- “Hochleistungsrechner sind Rechnersysteme, die geeignet sind, Aufgaben des Hochleistungsrechnens zu bearbeiten.”

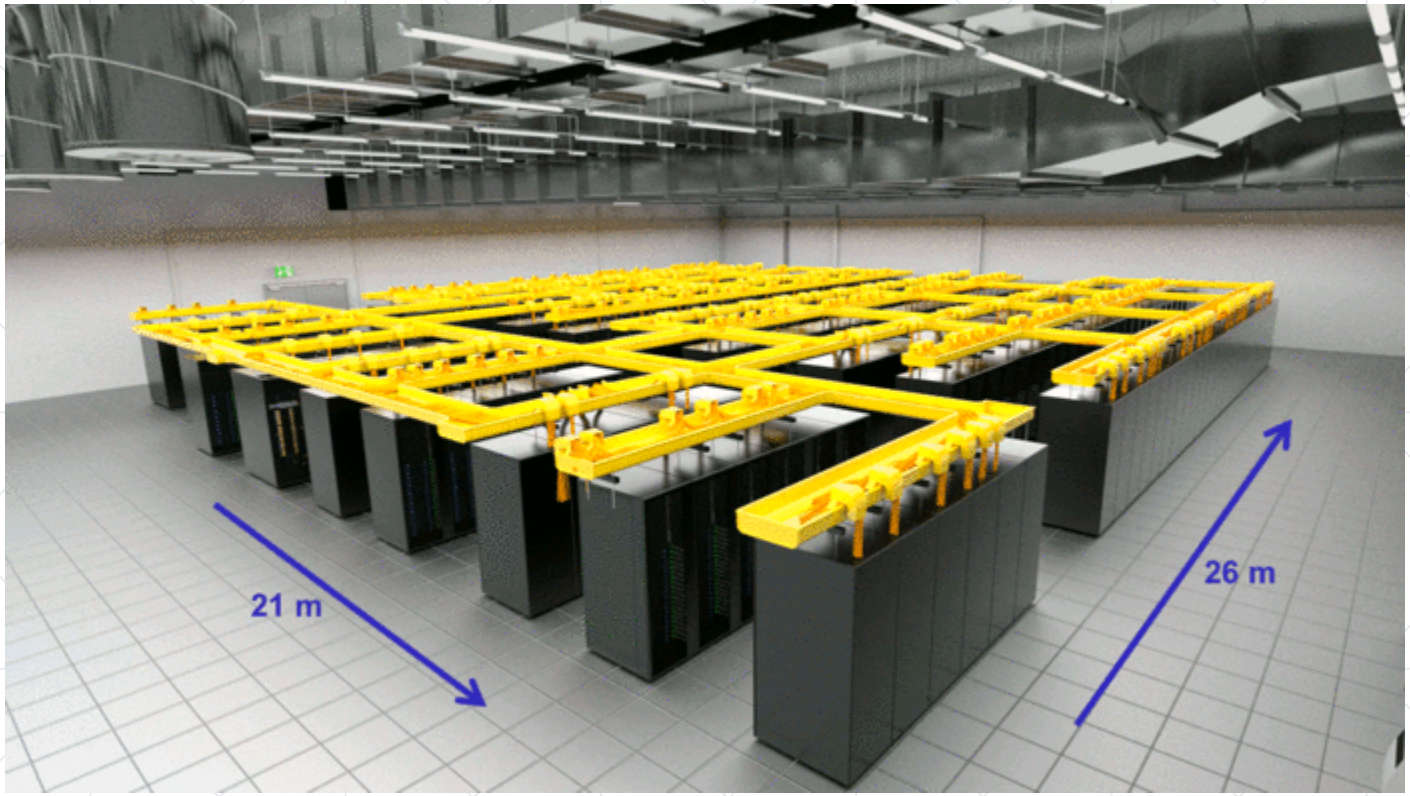
- Früher: Spezialsysteme – Spezialhardware + -software
- **Heute: Fast ausschließlich Cluster aus Standard-Hardware**, meist mit schnellem Interconnect (Netzwerk)
  - Die einzelnen Clusterknoten sind nicht schneller als ein gut ausgestatteter Standardserver!

- Vektorrechner: Nischenprodukt

Beispiel: SuperMUC Phase 1 (2012) @ LRZ

- **9216 compute nodes**
  - 2x Intel Xeon E5-2680 CPU (8 Kerne, 2.7 GHz, „Sandy Bridge“)
  - 32 GB RAM
  - Ergibt in Summe 147456 Kerne und 288 TB RAM
- Infiniband-Netz
  - 18 Inseln a 512 Knoten
    - Fully non-blocking innerhalb der Inseln
    - ausgedünnt 4:1 zwischen den Inseln
  - FDR-10 (knapp 40 Gbit/s)
- 15 PB paralleles Filesystem @250 GByte/s

“SuperMUC” Phase 1 (rendering)

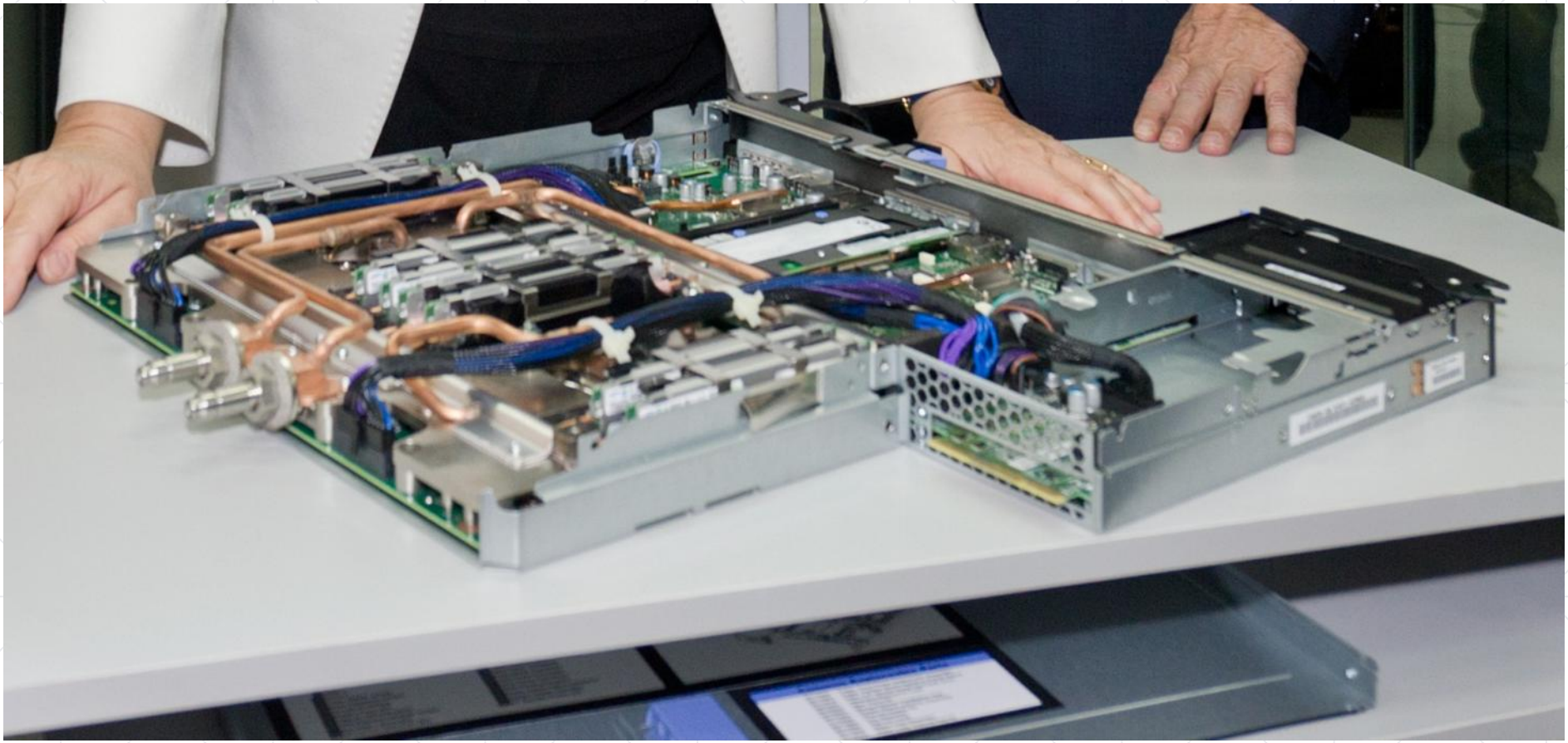


Quelle: www.lrz.de

SuperMUC Phase 1 (realer Kabelsalat)

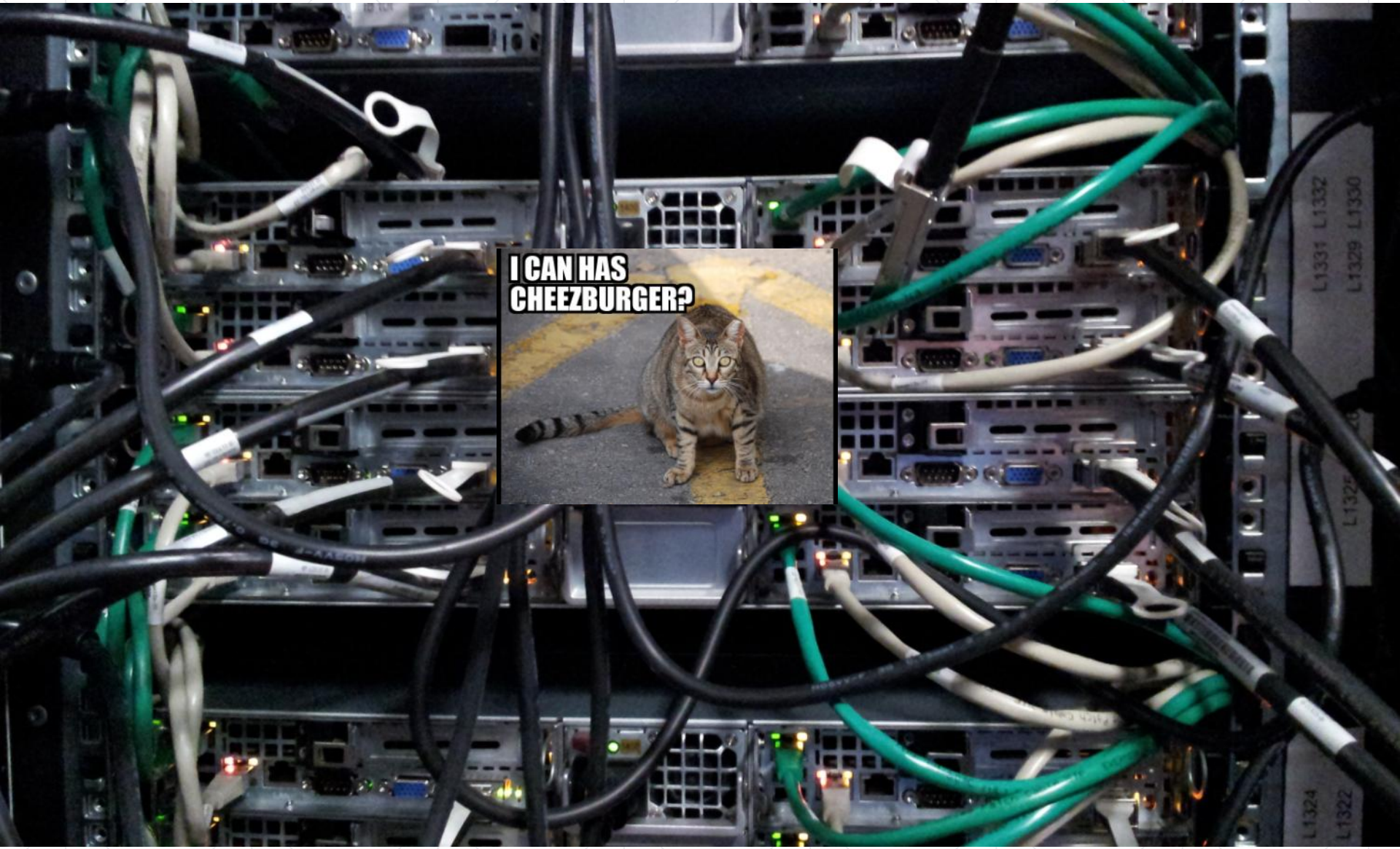
Quelle: www.lrz.de

SuperMUC (Rechenknoten mit Wasserkühlung)



Quelle: www.lrz.de

LiMa-Rechenknoten



Top500 http://www.top500.org

- Liste der **500 schnellsten HPC-Systeme** der Welt
  - …die öffentlich bekannt sein sollen
  - Gemessen mit "**Linpack**"-Benchmark
    (lösen eines großen linearen Gleichungssystems)
  - Erscheint 2x pro Jahr zu Konferenzen (meist November + Juni)
- **#1 November 2014: Tianhe-2, China**
- Trends aus der Liste vom November 2014
  - 45% mit Infiniband-Vernetzung
  - **>95% mit Linux (Windows: 0,2%)**
  - 45% in den USA (.de: 5%); Top10: 1 .cn, 6 USA, 1 .jp, 1 .ch, 1 .de
  - **85% mit Intel Xeon CPUs**

Infiniband

- **De-facto Standard** für HPC-Cluster
  - Außer Ethernet praktisch keine Konkurrenz mehr
- Schnell und mit sehr geringer Latenz
  - **Switched** fabric, oft fully non-blocking (bis 648 Ports)
  - QDR: 40 Gbit roh, 32 Gbit Nutz, 1.3 µs Latenz (seit 2007)
  - FDR: 56 Gbit roh, 54 Gbit Nutz, 0.7 µs Latenz (seit 2011)
- **Probleme**
  - Grauenhafte (Nicht-)API
  - Geringe Zuverlässigkeit
  - Besorgniserregende Marktkonsolidierung
    - Nur noch zwei Hersteller für QDR, nur einer für FDR
    - Stark verlangsamte Entwicklung, steigende Preise
    - „Jeder Depp denkt er könne ein IB-Netzwerk aufziehen“

Ein Teil des Infiniband-Netzes von LiMa…

Acceleratoren

- Einsteckkarten, auf die Berechnungen ausgelagert werden können
  - GPUs
  - Intel Xeon Phi

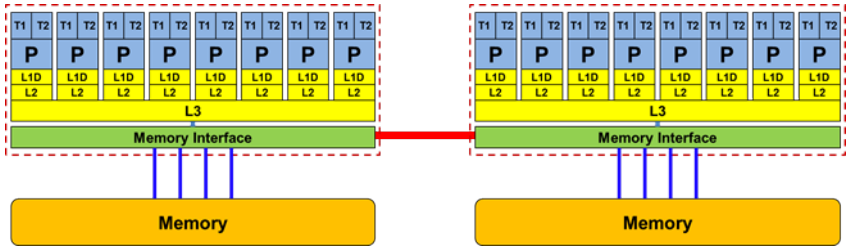
- Noch schwieriger effizient zu programmieren als normale shared-memory-Systeme oder Cluster. Programmiermodelle
  - NVIDIA CUDA, OpenCL (GPUs)
  - OpenMP (Intel Xeon Phi)
  - …
- **Wunder darf man auch im Idealfall nicht erwarten!**

Mit dem HPC-System läuft $GTA5 doch super, oder?

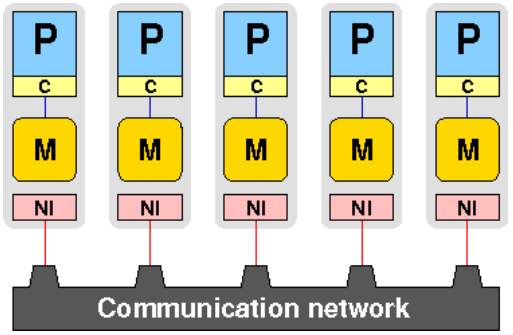
- HPC-Systeme führen nicht auf magische Weise beliebige Software schneller aus
- Software muss explizit dafür programmiert werden, dass sie das HPC-System nutzen kann
  - Parallelisierung
- → Selbst wenn das HPC-Cluster unter Windows laufen würde, könnte $GTA5 nicht mehr als einen Knoten eines Clusters benutzen.

Parallelisierungsarten

- Shared Memory
  - **OpenMP**: Compiler-Erweiterung
  - Pthreads: POSIX library
  - TBB, OmpSs, Cilk+,…
  - Automatische Parallelisierung: nicht praxisrelevant



- Distributed Memory
  - **MPI** (Message Passing Interface)
    - DER Standard
    - Version 1.0: Juni 1994
    - Aktuelle Version: MPI 3.1 (06/2015)



- Sprachbasierte Parallelität
  - X10, Chapel, …

OpenMP

- Spracherweiterungen, die von einem OpenMP-fähigen Compiler ausgewertet und umgesetzt werden
- Trivial-Beispiel aus Wikipedia (in C):

```
int i,a[100000];

#pragma omp parallel for
for (i = 0; i < N; i++)
    a[i] = 2 * i;
```

- Jeder Thread bekommt einen Teil der Schleife zugewiesen und arbeitet ihn ab (Shared Memory!)

MPI

- Alt aber konstant weiterentwickelt
- **Library-basiert** (Funktionsaufrufe)
  - Programmiersprache ist weiterhin seriell
- Praktisch jede Software die parallel auf mehr als einem Rechner laufen kann setzt darauf auf
- MPI-Standard definiert ein **Set von Funktionen** für die Programmierung von Parallelrechnern
  - Senden / Empfangen von Nachrichten
  - Parallel I/O
  - Synchronisation
    - › warten bis alle Prozesse eine Barriere erreicht haben
- Alle hochskalierbaren Produktionscodes auf Supercomputern sind MPI-basiert

“Hello World” in MPI (C)

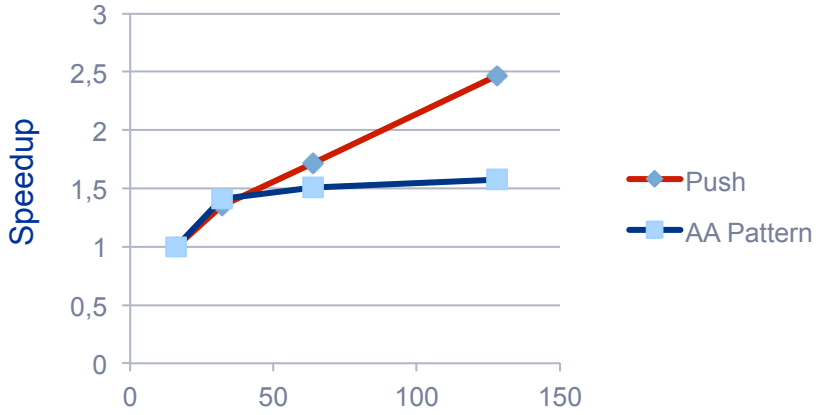
```
int main(int argc, char ** argv) {
 int numprocs, myid;

 MPI_Init(&argc,&argv);
 MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
 MPI_Comm_rank(MPI_COMM_WORLD, &myid);

 printf("Hi, ich bin Prozess %d von %d\n",
        myid, numprocs);

 MPI_Finalize();
 return 0;
}
```

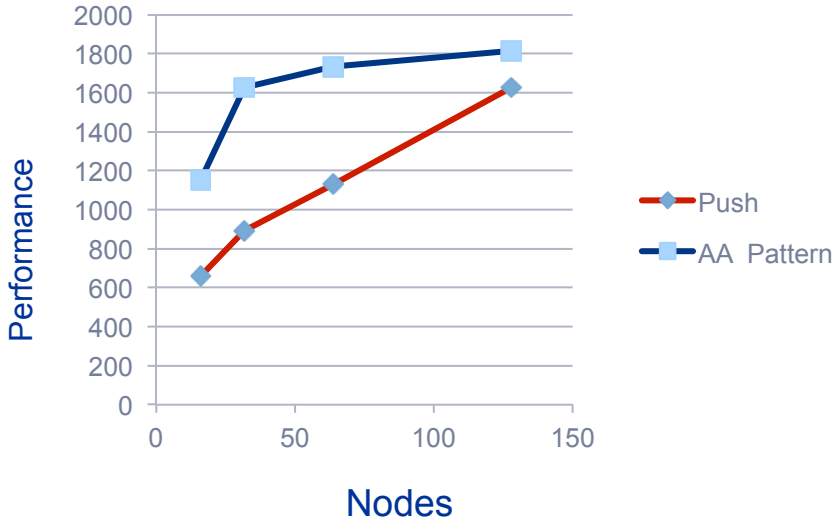
“Nun sag, wie hast du’s mit der Performance?”

- **Performance** = Leistung
- Leistung = **Arbeit / Zeit**
- **Was ist “Arbeit”** in der Computerei?
  - Populär im wissenschaftl. Rechnen: **Gleitkomma-Operationen (Flops)** **+ - *** (/)
  - Auch möglich: Spezifische Metriken (Pixels, Iterationen, Atome, Updates, Vergleiche, Vertauschungen, Schlüssel,…)
- Populärste Performancemetrik: **Flops/s** (GFlops/s, TFlops/s, …)
  - Peak Performance: Maximal mögliche Performance des Systems SuperMUC (Phase 1): $P_{peak} \approx 3$ PFlops/s
  - Real erreichbare Performance: Extrem abhängig von der Applikation (**durchschnittlich 3-10% von P_{peak}**)

Performance ≠ Speedup

- Performance = Arbeit / Zeit
- **Speedup** = „Wieviel mal schneller kann ich mit N CPUs rechnen als mit einer CPU?“
  - Speedup kann gut sein, während die Performance sehr schlecht ist
  - In der Tat ist das ein häufiges Zusammentreffen im HPC…





Nutzung von HPC-Clustern in der Praxis: Batchbetrieb

- Fast wie in der guten alten Zeit: Lochkarten mit Programmen und Daten darauf abgeben, auf Verarbeitung warten, Ergebnis abholen
  - Nur heutzutage natürlich ohne Lochkarten…
- Man kann sich nicht einfach beliebig auf beliebigen Knoten des Clusters einloggen

- Stattdessen: Einloggen auf **Frontend**, Jobs submittieren
  - Frontends sind interaktiv zugänglich (meist SSH)
  - Dort können Jobs vorbereitet und abgeschickt werden
  - Jobs müssen komplett ohne Benutzereingaben auskommen!

Beispiel-Batchscript

```
#!/bin/bash -l
#
#PBS -M michael.meier@fau.de
#PBS -m abe
#PBS -N LINPACK
#PBS -l walltime=01:45:00
#PBS -l nodes=16:ppn=24
#

cd $PBS_O_WORKDIR
module load intelmpi
mpirun_rrze -npernode 12 -pinexpr "S0:0-5@S1:0-5" ./xhpl
```

Spezifikation von Ressourcen und anderen Parametern

Arbeit!

Batchbetrieb

- **Scheduler** legt fest, wann der Job ausgeführt wird und berücksichtigt dabei Dinge wie…
  - **Priorität des Nutzers** und seiner Gruppe
  - wieviel Rechenzeit hat der Nutzer und seine Gruppe in der jüngeren Vergangenheit schon bekommen („Fair Share“)
  - Andere laufende Jobs
  - Reservierungen (z.B. für Nutzer oder Wartungsarbeiten)
  - Verfügbarkeit von Lizenzen die der Job braucht