



---

**NRW HPC Admin-Tag Dortmund**  
**Nutzung heterogener Ressourcen**

---

Dirk Ribbrock

**29.10.2019**

Fakultät für Mathematik und IT & Medien Centrum

## persönlicher Hintergrund

- Informatiker
- Anwendungsentwickler am Lehrstuhl für angewandte Mathematik und Numerik
- Admin bei Mathematik-Rechnerbetriebsgruppe und IT & Medien Centrum
- Clusternutzung:  
Dortmund (LiDO, Icarus), Münster, Köln, Jülich, Stuttgart, München, Erlangen,  
Barcelona, Los Alamos, Livermore, Penguin Computing, Amazon AWS

## inhärente Unterschiede zwischen Clustern/Rechenzentren

jeder Cluster besitzt eigene Hardware/Software-Ausstattung, die den Wechsel dorthin beeinflussen

- Code häufig nicht portabel zwischen Hardwareherstellern: NVIDIA vs. AMD vs. Intel, Intel vs. AMD vs. ARM vs. Power  
CUDA Code kann zum Gefängnis werden.  
Extrembeispiel: Codes für IBM Cell BE oder Intel Xeon Phi hatten unerwartet schnell keine Plattform mehr.
- Nutzer setzt Code ein, der nur mit exakt einer Compiler-Version funktioniert, oder zumindest nicht mit jedem Compiler
- langsame Umsetzung von Standards in 'neueren' Compilern
  - PGI vs. C++11
  - MS MPI vs. MPI 3
  - NVIDIA vs. OpenCL 2

## inhärente Unterschiede zwischen Clustern/Rechenzentren

jeder Cluster besitzt eigene Hardware/Software-Ausstattung, die den Wechsel dorthin beeinflussen

- verschiedene Scheduling-Systeme:
  - spätestens kompliziertere Skripte mit Abhängigkeiten häufig nicht direkt portierbar
- selbst beim gleichen Scheduling System werden unterschiedliche Befehl-Subsets unterstützt
- selbst in-place Tausch des Clusters erzwingt manchmal komplette Umstellung
  - Cray HazelHen → NEC Vulcan bedeutet Verlust aller Cray Tools und Einarbeiten in Ersatz

## beeinflussbare Faktoren

- Abweichungen in der Konfiguration:
  - queue/constraint Benennung, Einteilung, Regelung
  - insbesondere Behandlung von Beschleunigerhardware und großen Jobs  
zum Beispiel zusätzliches "gres=gpu:1" notwendig
  - kleinere Cluster häufig in sich heterogen und damit unübersichtlich  
Führt häufig zu suboptimaler Nutzung durch 'one size fits all' Lösungen
  - Storage-Konzepte: wo/wie/wer kann lesen/schreiben
    - LiDO: Jobs können nicht in home schreiben sondern nur in nobackup.
    - HLRS workspace mechanism.
  - Zugriff von/nach aussen, Zugriff innerhalb eines Clusters und eines Rechenzentrums zwischen Clustern.

## beeinflussbare Faktoren

- Umfang/Struktur der Dokumentation ist besonders Entscheidend bei Nutzung mehrerer Systeme
  - jedes Rechenzentrum nutzt eigene Struktur
  - häufig Einweg-Doku: einmal geschrieben, nie wieder angefasst
  - genauso häufig: copy & paste Doku vorheriger Systeme
  - Outsourcing bricht Struktur: Verweis auf Hersteller-Doku mit anderer Nomenklatur oder einfach tote Links.
  - Penguin verrät partout nicht, aus wievielen Knoten der Cluster überhaupt besteht.
  - Amazon benutzt völlig andere Begrifflichkeiten
- Viele Wege um Rechenzeit zu beantragen (Prace, Gauss, hpc.nrw, UA Ruhr, DoWiR) in verschiedenen Größenordnungen.

”Wieviel Rechenzeit und Ressourcen wird ihr Projekt in den nächsten 12 Monaten benötigen?!”

## Noch ein paar Beobachtungen zu Nutzern/Anwendern

- Publimachen der Option HPC (Profiling/Parallelisierung) in mehr Fachbereichen/Lehrstühlen.  
"Langsame Software ist ein Qualitätsmerkmal, denn was lange rechnet muss ja eine komplexe Rechnung sein."
- Für viele Anwender sind Struktur von Clustern, Unterschied Gateway/Nodes und Batch-Verarbeitung komplett neu.  
Auf dem Gateway rechnen kommt da schonmal vor.  
Auch beliebt: Immer nur ein Job gleichzeitig in der Queue.
- Häufig hat irgendwann ein Mitarbeiter alles eingerichtet und seitdem werden Scripte von einer Generation an die nächste weitergereicht ohne das noch jemand wüßte wie, warum und was die Scripte exakt machen.
- Senken der Einstiegshürde durch Integration der Cluster in den Desktop-Workflow (zB Matlab Parallel Server).

## Noch ein paar Gedanken zu Nutzern/Anwendern

- Ausbau des Monitoring um fehlerhafte Ressourcenallokation und ineffizientes Programmverhalten zu detektieren.  
Danach dann natürlich auch Hilfestellung beim Abstellen der Fehler.
- Benennung von Key-Usern in jedem Fachbereich/Standort, die als erste Ansprechpartner (vis-à-vis) zur Verfügung stehen.  
Häufig lassen sich 'Fehler' nur aufdecken, wenn ein geschultes Auge initiativ drauf schaut.
- Sensibilisierung für Ressourcenverbrauch (Energie / Personal).  
"Was spricht dagegen einen Cluster aus 2009 auch in 2019 noch weiter zu betreiben?  
Man spart sich doch die Umgewöhnung."
- Nutzer insgesamt bevormunden bzw. vor sich selbst schützen.  
CPU Frequenz absenken, Hyperthreading abstellen, Knoten nur exklusiv vergeben etc.



## **Vielen Dank fürs Zuhören**

Natürlich ist das meiste schon richtig gut gelöst, aber danach wurde ich nicht gefragt ;)