universität freiburg

Wissenschaftliches Rechnen und KI auf einer APU (AMD MI300/A)

Erste Erfahrungen im bwForCluster NEMO2

Universität Freiburg
Marko Glaubitz, Rob Falkenberg, Lars Pastewka, Bernd Wiebelt, Dirk von Suchodoletz

München, 09. September 2025

Ausgangslage: Beschaffung eines neuen Tier-3 Systems

Von NEMO1 zu NEMO2

- **bwForCluster NEMO**
  - bwForCluster NEMO („NEMO1“) in Betrieb seit 2016
  - Erneuerung und dabei Aktualisierung vieler Aspekte des Tier-3 Systems (bwHPC-Landeskonzept)
  - Formale Eröffnung von NEMO2 im September 2024 zum HPC-Symposium in Freiburg
  - Abschluss der Umbauten mit jüngst erfolgter KI-Erweiterung
- **Durchgehender Betrieb durch Art des Betriebsmodells**
  - Neue Knoten wurden im „NEMO1“ Modus betrieben
  - Umstellungen auf NEMO2 Modus dann sukzessive pro Rack, dadurch schrittweise Migration für die User (Umstellung MOAB auf Slurm, Umstellung BeeGFS auf WEKA)

Ausgangslage: Beschaffung eines neuen Tier-3 Systems

Überlegungen zu GPU

- **Beschaffung von NEMO1 damals ohne GPU**
- **NEMO1 wurde aber bereits zu einem frühen Zeitpunkt durch GPU erweitert**
  - Ein Server mit 8x NVIDIA V100 (Tesla; 32GByte)
  - Das geschah noch deutlich vor der KI-Revolution (Ergänzung im Zuge Neuberufung)
  - Die Hardware wurde nur zeitweise voll ausgelastet – dann hätte es aber mehr sein dürfen
- **GPU-Anteil für NEMO2 war gesetzt**
  - KI-Revolution zeichnete sich bereits im frühen Planungsstadium ab (aber vor ChatGPT4)
  - Forderung durch die Nutzer-Community für neue Forschungsansätze / -methoden
  - Frage: Welche GPU kaufen? NVIDIA/AMD/INTEL? Welche Modelle?

Gleiche Hardware (GPGPU), verschiedene Schwerpunkte

GPGPU – General Purpose Graphic Processing Unit

Ursprünglich: Grafik-Beschleunigung für Computerspiele. Inzwischen: Wesentlich mehr!

- **Wissenschaftliches Rechnen (WR / SC)**
  - 64bit Floating Point Operationen pro Sekunde (FLOPS)
- **Virtual Desktop Infrastructure (VDI), Virtual Classroom**
  - Grafikbeschleunigung (u.a. auch für CAD, Rendering, Videobearbeitung, ...)
  - Effiziente Videokodierung zur Übertragung auf entfernte Clients
- **Künstliche Intelligenz (KI / AI)**
  - Viele Operationen pro Sekunde (TOPs) mit kleinen Operanden (kleiner 64bit, teilweise kleiner 8bit)
  - Viel schneller Speicher auf der GPGPU für Trainings- und/oder Modelldaten (Minimum 32GB)

Eine GPGPU ist auf einen dieser Use-Case spezialisiert, kann (muss aber nicht) die anderen bedienen.

GPU: „Welches Schweinderl hätten’s den gern?

Oder „choose your poison“

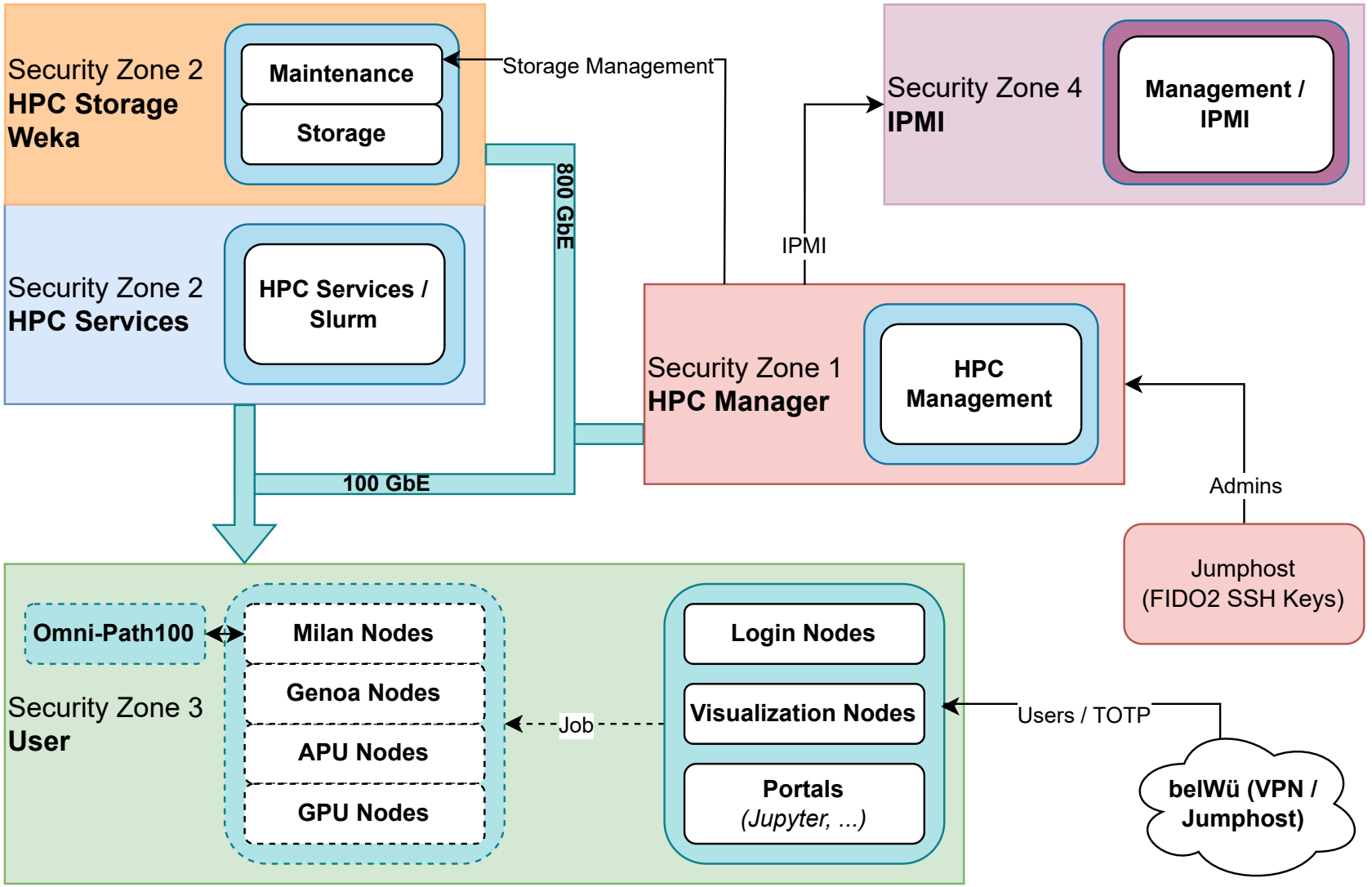
- NVIDIA ist klar der Platzhirsch, mit allen Vor- und Nachteilen
  - CUDA teilweise harte Voraussetzung bzw. „works out of the box“
  - Teuer und selbstherrliches Auftreten. Erinnert an Intel vor 20 Jahren
- AMD mit kompetitiver Hardware, aber auch mit Problemen
  - Software-Stack ist eher was für Bastler; Kontakt zu bekommen - mühsam
  - APU (MI300/A) mit shared CPU/GPU Memory als neues Konzept will erst mal verstanden werden
- INTEL momentan nicht konkurrenzfähig
  - Beim Treiber-Support durchaus engagiert (Spezialfall VDI)

GPU: Welches Schweinderl wurde es denn?

Hint: „Gemischtwarenladen“

- 9 Nodes mit 4x NVIDIA L40S (48 GByte)
  - Für „kleine, reine“ KI-Bedarfe (gegenüber L40 keine nennenswerte Float-Performance)
  - Argument: CUDA; Schnellstart mit bekanntem Code
- 4 Nodes mit 4x AMD MI300/A mit kompetitiver Hardware, aber auch mit Problemen
  - Dual Use: Wissenschaftliches Rechnen und KI, in höheren Tiers in Verwendung (HLRS)
  - Software-Stack ist eher was für Bastler, Besserung wurde immerhin versprochen
  - APU (MI300/A) als neues Konzept will erst mal verstanden werden
- 2 Nodes mit 8x NVIDIA H200 (141 GByte; aus zusätzlicher MWK-Förderung für KI)
  - Als Komplement zu den Nodes mit MI300/A, im Prinzip rein für KI-Aufgaben

NEMO2 Aufbau



Exkurs 1: Energieeffizienz

Energieeffizenz als Schwerpunkt, aber KI ändert die Spielregeln

- Beschaffung von NEMO2 mit klarem Fokus auf Energieeffizienz bei der CPU-Partition
  - Induziert auch wegen Beschränkung durch Infrastruktur (Kälteversorgung, nicht Stromversorgung)
  - Natürlich auch: Klimaschutz und stark gestiegene Energiekosten
  - Ausschreibung deshalb mit dem Versuch auf Performance pro Watt zu optimieren (x86-64 noch als Architektur gesetzt), siehe dazu auch den Talk am Nachmittag
- Bei der GPU-Partition sind solche Energieeffizienz-Kritierien schlicht (noch) nicht marktwirksam
  - Die Diskussion über Effizienz hat erst begonnen
  - Derzeit läuft noch das Wettrennen um die höchste KI-Leistung (absolut, nicht pro kW)

Exkurs 2: Virtual Desktop Infrastructures / Virtual GPUs

The lost (use-) case: Virtualisierung und Visualisierung

- Mit hardware-unterstütztem Rendering und Video-Stream en- und de-coding für schnellen Transport hoher Auflösungen
- Keine dedizierten Server mehr für diese Zwecke sondern VMs bzw. Container auf einem Knoten zusammengefasst, Aufteilung einer GPU in virtuelle GPUs mit eigener PCI-ID (SRIOV)
- Zusätzlich in Zukunft: Full Desktop Remote Access auf Basis einer Open Source Virtual Desktop Infrastructure für breiteres Angebot von Betriebssystemen und Applikationen
- Herausforderung: Geeignete GPU
  - Intel mit Flex-Server GPU (140/170) für VDI, flexible Partitionierung, gute Treiberunterstützung
  - NVIDIA mit aufgepropftem Lizenzmanagement für GPU Virtualisierung
  - AMD mit Radeon Pro theoretisch Option, aber bisher unmöglich diese zu beschaffen

AMD MI300A

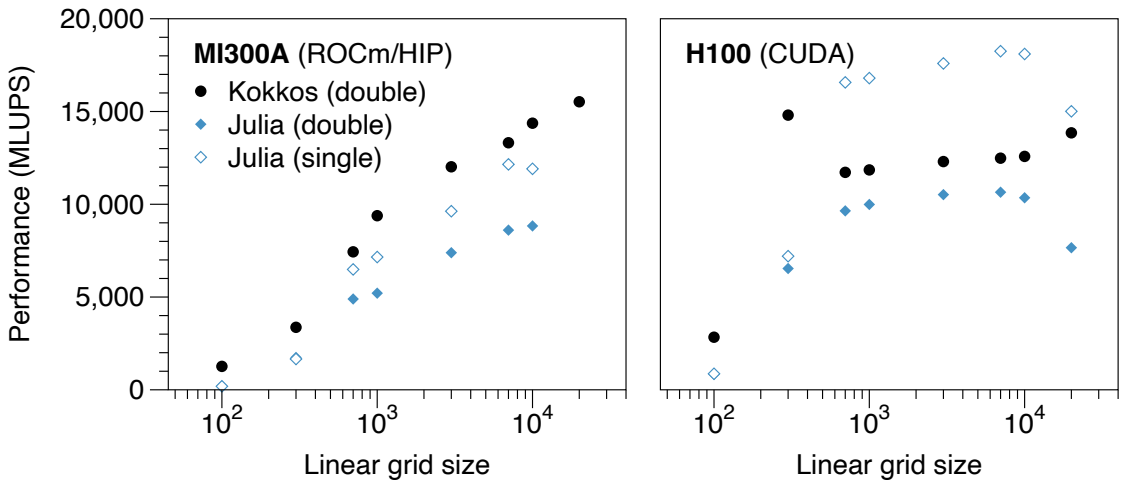
Überlegungen zur Konfiguration

- Im SXM Faktor (Bauform) als APU (also mit gemeinsamen HBM3-Speicher für GPU und CPU)
- Nodes mit 4x MI300A SXM Modul als „Sweet Spot“
- 128 GB HBM3 Speicher pro SXM Modul, 24 CPU Cores pro SXM-Modul
- Somit 512 GB schneller HBM3 Speicher und 96 CPU Cores gesamt pro Node
- Für Dual-Use (HPC und KI) konzipiert mit FP64 Performance für wissenschaftliches Rechnen und 512 GB unified HBM3-Memory für LLM, ohne PCI-Express-Bremse
- Fokus liegt auf Nutzung von nativen Bibliotheken für TensorFlow und PyTorch
  - CUDA-Kompatibilität durch Emulation spielte in den Überlegungen keine Rolle

AMD MI300A – Use Case „Wissenschaftliches Rechnen“ (1)

Hier am Beispiel von LAMPS aus der Mikrosystemtechniken

- 2D Lattice Boltzmann Rechnung, auf der y-Achse ist die Performance (mehr ist besser). x-Achse ist eine Seitenlänge des Gitters, also 1000 wäre ein 1000 x 1000 Gitter mit 10^6 Gitterpunkten. Die x-Achse hört da auf, wo der Speicher der Karten voll war.
- Die Performance der MI300A ist hier vergleichbar mit der H100. Warum die Performance auf der MI300A mit der Gittergröße so ansteigt – unklar.
- Kein signifikanter Geschwindigkeitsgewinn bei Nutzung von Single Precision statt Double Precision.



AMD MI300A – Use Case „Wissenschaftliches Rechnen“ (2)

Hier am Beispiel von LAMPS aus der Mikrosystemtechnik

- Direkte Nutzung der MI300A über ROCm/HIP eher sekundär für die Arbeitsgruppe. Ebenso CUDA.
- Abstraktionsschichten für NVIDIA und AMD:
  - Kokkos: C++ Bibliothek, die Performance auf den H100 ist nahe dran an einer reinen CUDA Implementierung. Sieht aus Sicht der Arbeitsgruppe sehr gut aus.
  - Julia: 10-30% geringere Performance respektive Kokkos, allerdings deutlich einfacher zu programmieren. Es ist anzunehmen, dass die Performance mit neueren Versionen der LLVM Compiler eher besser wird
  - Taichi: Wie Julia LLVM basiert, aber Python Syntax. Sah mit NVIDIA Karten auch sehr gut aus.

AMD MI300A – Use Case „Wissenschaftliches Rechnen“ (3)

Hier am Beispiel von LAMPS aus der Mikrosystemtechniken

- Vergleich - für 15.000 MLUPS (die sowohl 1x MI300A als auch 1x H100 liefern) hätte die Arbeitsgruppe Größenordnung 3.000 Kerne auf NEMO1 gebraucht
- GPU können wissenschaftlichen Output per KWh verbessern, so Expertise und Programmierfähigkeiten vorliegen
- Im Moment noch schwierig Nutzer zu begeistern
  - Wobei: Aussicht auf ungestörte Nutzung exklusiver Ressourcen
- Typisches Problem/TODO: “Habe FORTRAN coarrays ausprobiert, aber nicht auf den GPU zum Laufen gebracht“

AMD MI300A – Use Case „Künstliche Intelligenz“ (1)

Neben HPC: Entwicklung einer KI-Strategie für Campus

- Wegen verhaltener Nutzung eine MI300A-Box an E-Learning-Gruppe für Tests „verliehen“ (ebenso klassische CPU-Boxen mit 2* NVidia L40S wegen langsam anlaufender Nutzung)
- Motivierte Testnutzer – hatten den Auftrag Erfahrungen zurückzumelden
  - ollama
  - vLLM
  - L40S vs. MI300A (preliminary only)

AMD MI300A – Use Case „Künstliche Intelligenz“ (2)

Tests: Modell ollama

- Laut Doku Software mit der MI300A kompatibel (siehe GitHub zu ollama)
- War sie wohl während verschiedener Tests der Kollegen nie (1. Halbjahr 2025)
  - Mehrmaliges Kompilieren unterschiedlicher Branches, Fixes und Versionen brachte keine Ergebnisse, vgl. u.a.: GitHub-PullRequest
- Teilweise in die Tiefen von ROCm (Open software stack that includes programming models, tools, compilers, libraries, and runtimes for AI and HPC solution development on AMD GPUs) abgestiegen (eher relativ komplex und am Ende wenig fruchtbar)
- In der Zwischenzeit Umstieg auf vLLM, da deutlich breiteren und besser konfigurierbaren Funktionsumfang (insbesondere auch im Hinblick auf skalierbaren Betrieb)
- NVIDIA Grafikkarten übrigens sehr problemlos bespielbar mit ollama (L40S)

AMD MI300A – Use Case „Künstliche Intelligenz“ (3)

Tests: Modell vLLM (Berkeley/Community optimiertes LLM)

- Relativ einfach skalierbar über mehrere GPUs inklusive Eingabe eines Kontextfensters
- Wie immer L40S + RTX 2080 recht schnell am Start (NVidia), während mit MI300A mehr Aufwand
- Berichtsstand Juni 2025: (2 Monate mittlerweile leider literally "alt"): Notwendige Docker Image inklusive der notwendigen ROCm Treiber per Hand zu bauen: vgl. vLLM
- Für Neuen gpt-oss Modelle vorgefertigte Docker-images (vgl. Link)
- gpt-oss:120b mit erheblichen Problemen beim Starten. Bisher intern nicht gelöst
- gpt-oss:20b läuft relativ direkt out of the box. (d.h. 1.5h Aufwand bis zur Sicherstellung der Grundfunktionalität

AMD MI300A – Use Case „Künstliche Intelligenz“ (4)

Messungen / Versuch von Vergleichen: Token/s L40S und MI300A

- Achtung: Keine validen Messungen, sondern nur Stichproben mit folgender Anfrage: "Weißt du wie ich sehe, wie viele Tokens pro Sekunde ein Large Language Modell verarbeitet?":
- L40S mit gpt-oss:20b: `(APIServer pid=1) INFO 08-25 07:11:47 [loggers.py:123] Engine 000: Avg prompt throughput: 9.1 tokens/s, Avg generation throughput: 32.3 tokens/s, Running: 1 reqs, Waiting: 0 reqs, GPU KV cache usage: 0.2%, Prefix cache hit rate: 64.3%`
- Fazit: MI300A scheint für einzelne Prompts etwas schneller als L40S
- Allerdings MI300A komplett unkonfiguriert ohne Beachtung von
  - möglicher Kontext Size
  - Parallele Requests nicht getestet
  - unklar, ob dieselben Quantifizierungen der Modelle geladen wurden

AMD MI300A – Use Case „Künstliche Intelligenz“ (5)

Weitere Schritte / ToDos / Fazit:

- Modelle über mehrere MI300A verteilen
- Mehrere Modelle auf einer MI300A parallel laufen lassen
- Präzisere Messungen von Output
- Unterschiedliche Quantisierungen ausprobieren und vergleichen
- Parameter-Extraktion über Monitoring

- Fazit: Bleibt weiter spannend – Ziel sich breit aufzustellen, jedoch etliche Hürden
- Am Ende wieder der gute alte Kapitalismus: Kapital kann durch Arbeit substituiert werden (und umgekehrt)