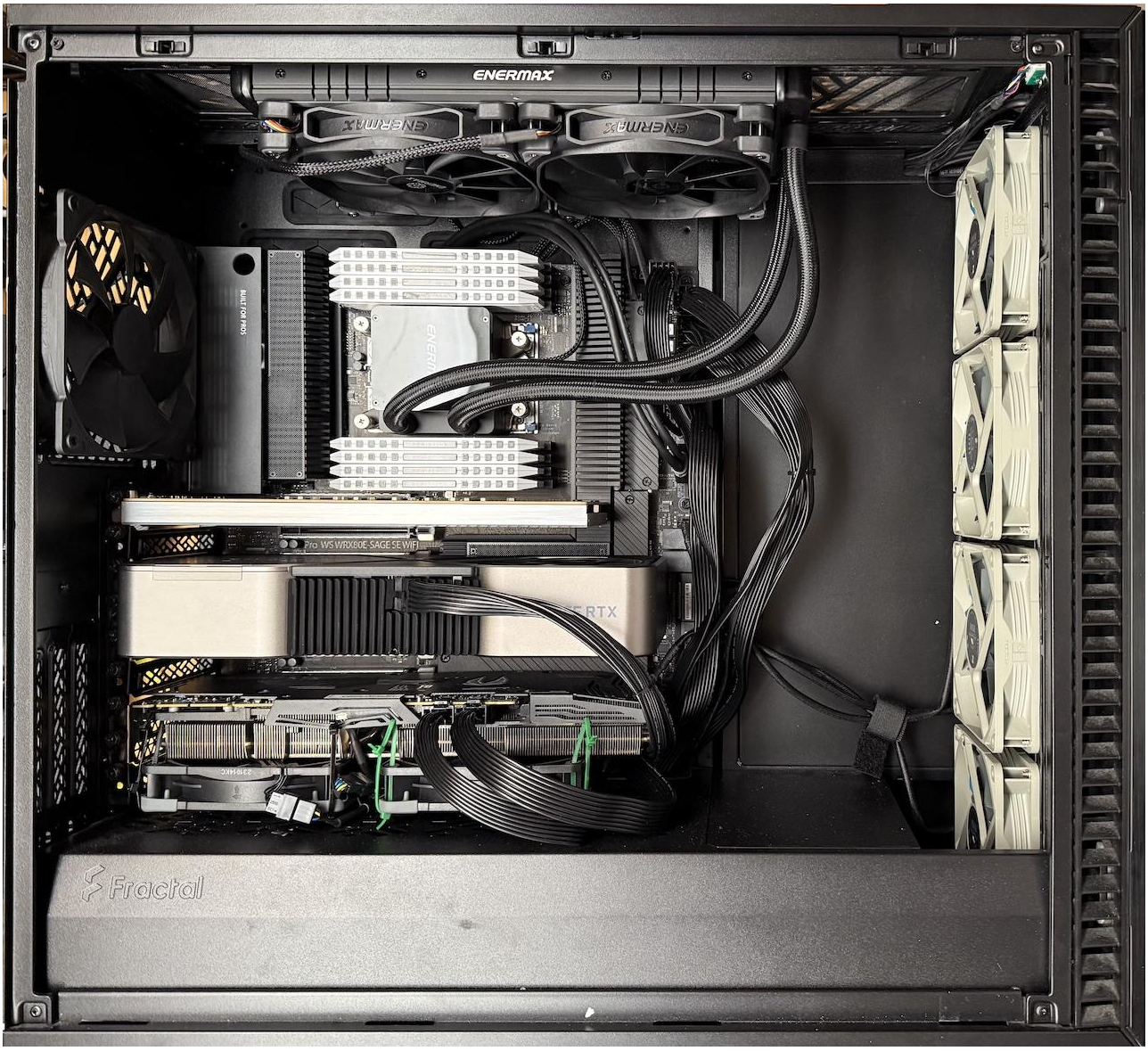
KI-Server

PING hat 2025 einen KI-Server angeschafft. Er heißt **cogito.ping.de** und befindet sich im Rechnerraum des Gebäudes in der Joseph-von-Fraunhofer-Straße.

Photo

Und so sieht das Innenleben aus: Ganz oben hängt der Kühlkörper für die CPU-AIO-Wasserkühlung. Ober- und unterhalb der CPU jeweils 4 weiße RAM-Module, darunter im 1. Slot die 4x NVMe-SSD-Karte und darunter die aktuell zwei RTX 3090 GPUs. Das Netzteil ganz unten liegt verborgen unter der Abdeckung, zusammen mit einem weiteren 140mm Lüfter. Rechts im Bild sorgen 4 Noctua-Lüfter für Durchzug.



Technische Daten

- CPU AMD Threadripper Pro 5955WX 16 cores 32 threads 4.5Ghz, Zen 3, 64MB L3, 128 PCIe 4.0 lanes
- Mainboard Asus Pro WS WRX80E Sage SE Wifi (7x PCIe 4.0 x16, 8x DDR4 DIMM, 3x M.2, …)
- 2x RAM Corsair Dominator Platinum RGB White UDIMM 64GB **Kit** DDR4-3600 CL18-19-19-39 (128GB gesamt)
- GPU NVIDIA GeForce RTX 3090 Founders Edition 24 GB
- GPU Zotac Gaming GeForce RTX 3090 Trinity OC 24 GB mit Noctua Lüftern
- Fractal Design Define 7 XL Black TG Dark Tint schallgedämmt Big-Tower (12 PCI Steckplätze)
- SilverStone IceGem 240P AIO CPU-Wasserkühlung
- Antec Neo Eco Gold Modular NE1300G m 1300W ATX 3.0 Netzteil
- 4x SSD Lexar NM790 1TB M.2 NVMe PCIe 4.0 in Asus Hyper M.2 X16 Gen 4 Card (RAID 0)
- SSD Samsung PM951 512GB M.2 NVMe (boot)
- SSD Samsung EVO 850 500GB S-ATA
- 4x Noctua NF-P12 redux-1700 PWM 120mm Lüfter

siehe auch https://geizhals.de/wishlists/3870524

Zu dem Mainboard gehört auch eine PCIe 4.0 x16 Karte um vier PCIe 4.0 x4 NVMe SSDs anzuschließen. Dort befinden sich die 4 Lexar SSDs.

Durch die 2 GPUs stehen derzeit 48GB schnelles VRAM zur Verfügung, eine Erweiterung ist möglich. Ins Gehäuse passen maximal fünf 2-slot GPUs.

Der Hauptspeicher ist auf acht 16GB-Module verteilt und nutzt so die 8 Speicherkanäle der AMD Threadripper Pro Architektur.

Der Platz im Gehäuse und das Mainboard mit vielen PCIe-Lanes ermöglichen es uns, bei Bedarf noch mehr GPUs einzubauen.

Netzwerk

Der Rechner ist im PING Büronetzwerk, er ist nicht direkt aus dem Internet erreichbar. Auf milla läuft ein nginx der Zugriffs an open-webui und die vLLM API weiterleitet.

Im Büro-LAN hat cogito.buero.ping.de die IP 192.168.200.245 und das Management Interface ist auf 192.168.200.240 erreichbar.

Software

Auf dem Rechner läuft derzeit Ubuntu 24.04 LTS.

Das Konzept sieht vor, die Dienste als Docker Container laufen zu lassen. Die compose Dateien liegen in Unterverzeichnissen unter `/opt`.

Mit `nvtop` kann der Status der GPUs angezeigt werden.

Über das Script `/usr/local/bin/set-gpu-power.sh` sind die GPUs aktuell auf 280W gedrosselt. Das führt zu nur sehr geringen Leistungseinbußen.

NVidia Kernel Modul

Nach Installation eines neuen Kernels müssem auch die entsprechenden Header installiert werden, damit das Nvidia Kernel Modul kompiliert wird:

```
sudo dpkg-reconfigure nvidia-kernel-common-580
dkms autoinstall
apt install linux-headers-6.17.0-1021-nvidia
modprobe nvidia
```

(Hier am Beispiel von Linux 6.17.0-1021).

Danach ist leider noch ein Reboot vonnöten.

Storage

Die 4 Lexar NMVe SSDs bilden ein RAID0 das unter `/opt` eingehängt ist. Dadurch können große LLMs schnell in den Speicher geladen werden.

Die S-ATA SSD ist unter `/var/local` eingehängt.

Die PM951 NVMe SSD dient als Bootlaufwerk und enthält auch das OS.

```
$ lsblk
NAME                      MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINTS
sda                         8:0    0 465.8G  0 disk
```

```
└─sda1                      8:1    0 465.8G  0 part  /var/local
nvme0n1                   259:0    0 953.9G  0 disk
└─md0                       9:0    0   3.7T  0 raid0 /opt
nvme1n1                   259:1    0 953.9G  0 disk
└─md0                       9:0    0   3.7T  0 raid0 /opt
nvme3n1                   259:2    0 953.9G  0 disk
└─md0                       9:0    0   3.7T  0 raid0 /opt
nvme2n1                   259:3    0 953.9G  0 disk
└─md0                       9:0    0   3.7T  0 raid0 /opt
nvme4n1                   259:8    0 476.9G  0 disk
├─nvme4n1p1               259:9    0     1G  0 part  /boot/efi
├─nvme4n1p2               259:10   0     2G  0 part  /boot
└─nvme4n1p3               259:11   0 473.9G  0 part
  └─ubuntu--vg-ubuntu--lv 252:0    0   445G  0 lvm   /

$ blkid
/dev/nvme0n1: UUID="ebe75b1c-af8f-5e3a-aa0f-9464c3951451" UUID_SUB="1d5de863-0634-7dd3-0e97-7dec
/dev/nvme3n1: UUID="ebe75b1c-af8f-5e3a-aa0f-9464c3951451" UUID_SUB="440d89df-a9d1-bfc9-3634-e57b
/dev/md0: LABEL="RAID" UUID="f57d1a53-8b0c-4119-a02b-e06632c7933d" BLOCK_SIZE="4096" TYPE="ext4"
/dev/nvme2n1: UUID="ebe75b1c-af8f-5e3a-aa0f-9464c3951451" UUID_SUB="eb15293d-d63b-c940-841d-a918
/dev/mapper/ubuntu--vg-ubuntu--lv: UUID="98bd9894-3827-42bb-a0f4-d92931530cab" BLOCK_SIZE="4096"
/dev/nvme1n1: UUID="ebe75b1c-af8f-5e3a-aa0f-9464c3951451" UUID_SUB="797f9137-3a34-2689-5d0b-a294
/dev/sda1: LABEL="ssd500" UUID="c57d324d-3c4f-4f5d-90ca-3859ca87f550" BLOCK_SIZE="4096" TYPE="ex
/dev/nvme4n1p3: UUID="XbVKNc-zwqt-qe2c-fj2e-8MRA-p8e0-XDQdsz" TYPE="LVM2_member" PARTUUID="5f653
/dev/nvme4n1p1: UUID="7006-F657" BLOCK_SIZE="512" TYPE="vfat" PARTUUID="f57a2cbc-da3a-4322-8921-
/dev/nvme4n1p2: UUID="8256bdab-088d-437e-a82b-b94470729f4c" BLOCK_SIZE="4096" TYPE="ext4" PARTUU

$ cat /proc/mdstat
Personalities : [raid0] [linear] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid0 nvme3n1[3] nvme2n1[2] nvme0n1[0] nvme1n1[1]
      4000288768 blocks super 1.2 512k chunks

unused devices: <none>
```

vLLM mit Open-WebUI

Für Inferenz läuft i.d.R ein vLLM Server. Als WebUI gibt es dafür ein open-webui.

Für das Umwandeln von Office Dokumenten (zum Beispiel ODT) läuft Apache Tika.

Das Docker compose file liegt unter `/opt/vllm/`

Auf milla.ping.de (aka buero) läuft ein nginx der open-webui unter https://ki.ping.de erreichbar macht. Für den Login nutzt bitte unser Single Sign-On.

Der vLLM bietet auch direk auf Port 8000 unter https://ki.ping.de:8000/v1 eine OpenAI-kompatible API. Ihr benötigt das Bearer Token, ihr erhaltet es auf dieser geschützten Wiki-Seite.

Ollama mit Open-WebUI

Alternativ können wir den Ollama Server starten. Der ist aber nicht mehr der bevorzugte Dienst, u.a. weil die Performance mit mehreren Usern schlecht ist.

Das Docker compose file liegt unter `/opt/ollama/`

Beim Einsatz von Ollama erscheint im Model-Selektor von Open-WebUI ein grüner Punkt neben den LLMs, die derzeit im GPU Speicher sind.

Das Script `/usr/local/bin/ollama-nogpu.sh` ist dafür da, den Ollama Container neu zu starten, falls dieser mal wieder die GPUs nicht erkennt.

ComfyUI

ComfyUI (primär für KI-Bildergenerierung) ist noch nicht fertig installiert, es liegt unter `/opt/comfyui` und kann bei Bedarf gestartet werden. Vorher sollte ollama gestoppt werden, weil nicht genügend GPU VRAM für beide Dienste gleichzeitig vorhanden ist.

Revision #16
Created 2026-01-31 17:00:08 UTC by Daniel Hess
Updated 2026-10-18 11:29:35 UTC by Sven Neuhaus