OpenGPT-X

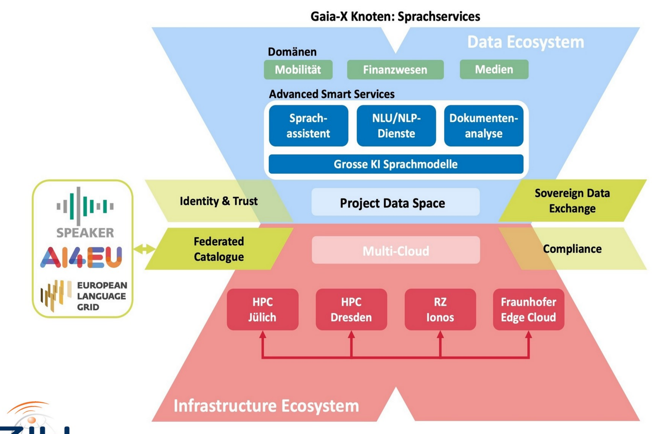
Training Large Language Models on HPC Systems

Carolin Penke, Chelsea John, Andreas Herten, Jan Ebert, Stefan Kesselheim, Estela Suarez

OpenGPT-X

Supported by: Federal Ministry for Economic Affairs and Climate Action, on the basis of a decision by the German Bundestag

- Since 2022 OpenGPT-X builds and trains **large-scale AI language models** to drive innovative **language application services** for the **European economy.**
- Embedded in the **Gaia-X** infrastructure
- 11 partners from **academia** and **industry**
- **Pilot use cases** included



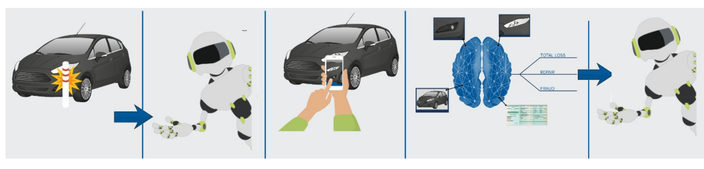
Large Language Models and Applications

A language model is a probability distribution predicting the next word in a sentence:

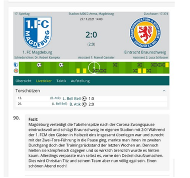
$$P(w_t|w_{1:(t-1)}), \qquad w_1, \dots, w_{t-1}, w_t \in V.$$

Use Case Examples

1) Digital assistant for customer claims journey



2) Writing sports match reports



...

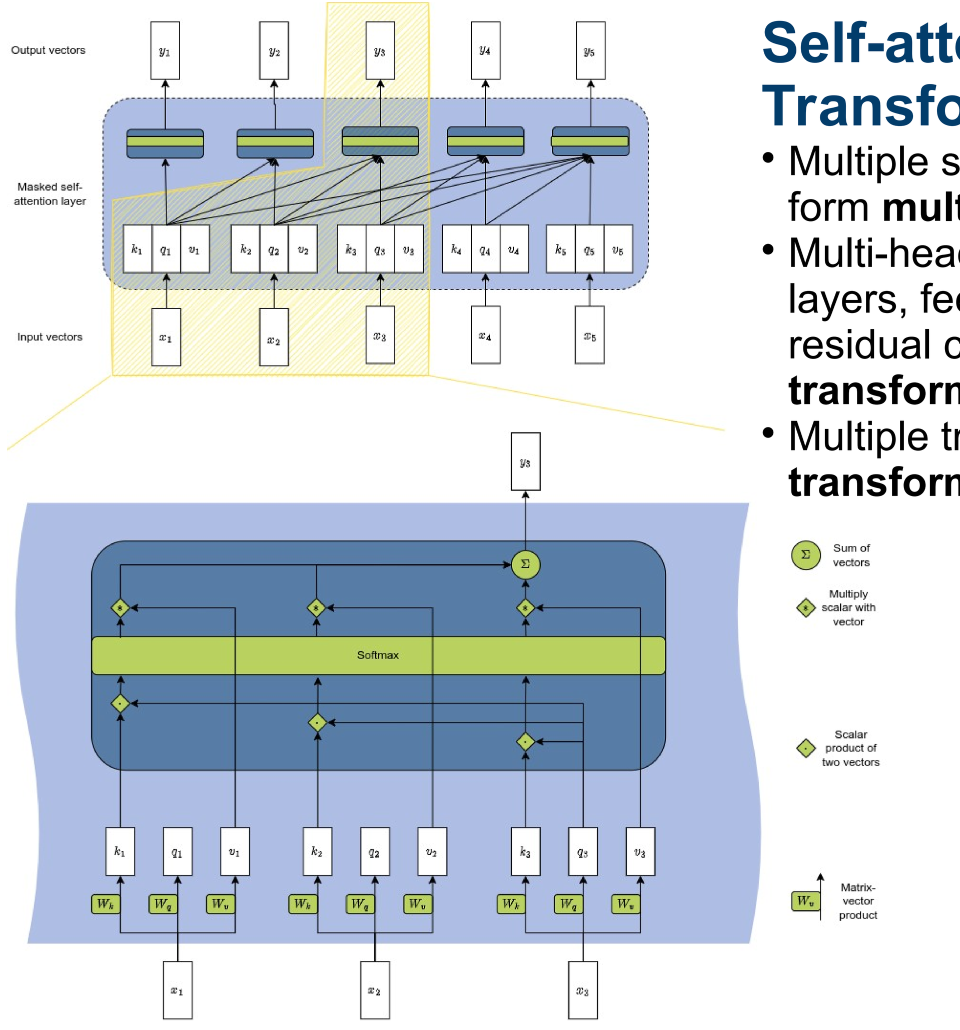
Other Language Models

GPT, *Generative Pre-trained Transformer* by OpenAI, GPT-1 (2018), GPT-2 (2019), GPT-3 (2020), (available for fee)
OPT, *Open Pre-trained Transformer* (2022), by Meta (available to researchers for free)
BLOOM (2022), by BigScience (HuggingFace), based on Megatron-Deepspeed
Megatron: Open-Source framework for training transformers at scale, by Nvidia
Deepspeed: Distributed training library by Microsoft, implement ZeRO stages

Language models need to be trained with lots of data on supercomputers!

The Transformer Architecture

Recent Breakthroughs possible because of novel network architecture called transformer, based on **self-attention layers**:



Self-attention Layers form Transformer Architectures

- Multiple self-attention layers in parallel form **multi-headed attention**.
- Multi-headed attention, normalization layers, feed-forward layers, and residual connections form a **transformer block**
- Multiple transformer blocks form a **transformer**.

Advantages over RNN and LSTM Architectures

+ mediates **vanishing gradient effect**
+ not inherently sequential, **parallel computations** possible
+ matrix-**matrix products**

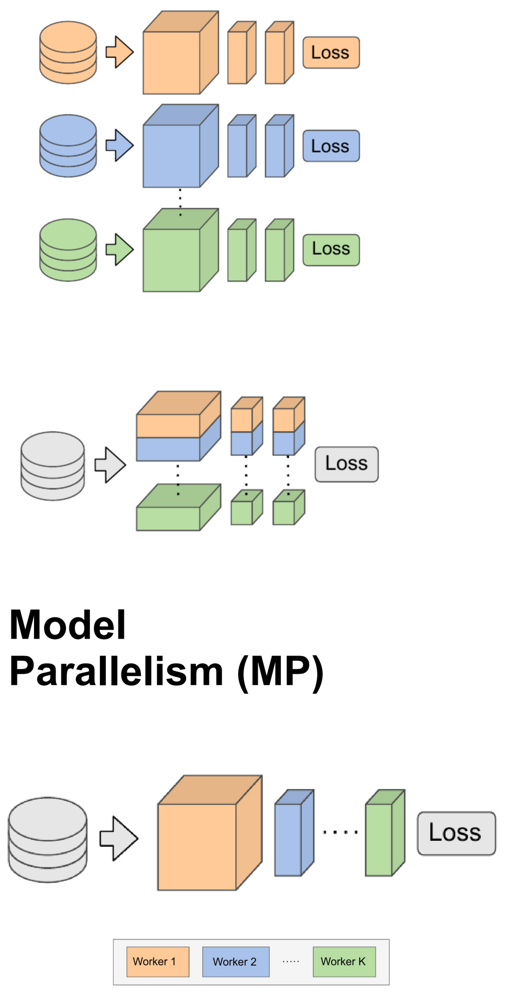
C. Penke, 2022, *A mathematician's introduction to transformers and large language models*, JSC Accelerating Devices Lab Blog, https://doi.org/10.34732/xdvblg-qsbtyx

3D Parallelism for Training

To scale to a full supercomputer three kinds of parallelism are intertwined.

1. **Data parallelism (DP)**
   - Input data is distributed across ranks
   - Full model replica on each rank
   - Gradients are computed for local mini batch
   - Gradients are synchronized during backward propagation (all-reduce)
2. **Pipeline parallelism (PP)**
   - Layers of each data parallel replica are distributed across ranks
   - Overlay computations by dividing local mini batch into micro-batches (Gradient accumulation steps)
   - Clever scheduling strategies
3. **Tensor parallelism (TP)**
   - Weight tensors (matrices) of each pipeline stage are distributed across ranks
   - Multi-headed attention parallel by nature
   - Feed-forward layers distributed column- or row-wise to minimize communication
   - Communicaiton intensive (two all reduces per layer), NVLink useful

Model Parallelism (MP)



Parallel Training of Deep Networks with Local Updates, Laskin, Metz, Nabarro, Saroufim, Noune, Luschi, Sohl-Dickstein, Abbeel, 2020

#GPUs = DP × PP × TP = DP × MP

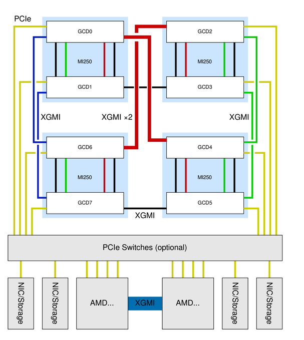
Novel Architectures

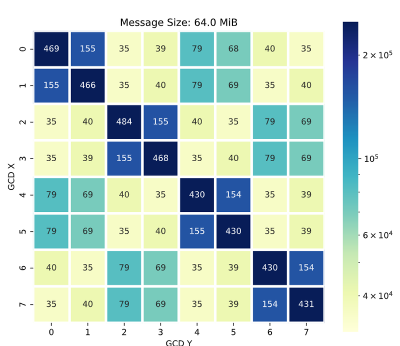
Flagship Cluster: Juwels Booster with> 3200 Nvidia A100 GPUs, 40 GB

JURECA Evaluation Platform

- additional nodes for evaluation and testing

AMD Instinct MI250 GPUs





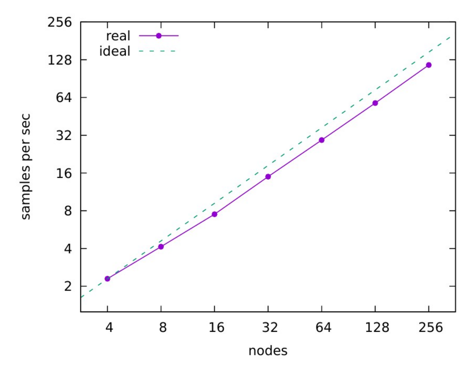
A. Herten, 2022, *First Benchmarks with AMD Instinct MI250 GPUs at JSC*, JSC Accelerating Devices Lab Blog, https://doi.org/10.34732/xdvblg-rmlyc3

Graphcore IPU-POD4

https://docs.graphcore.ai/projects/graphcore-ipu-m2000-datasheet/en/latest/index.html

Scalability and Model Layouting

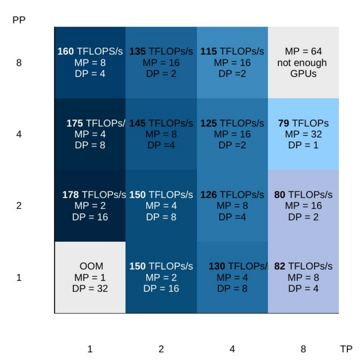
- Training highly scalable
- Goal: **High throughput** on small node counts (for testing) and large node counts (final training).
- ~ **50 % of peak** e (312 TFLOPS/s) easily possible
- Good performance and simplicity: Data parallelism, limited scalability
- Problem: Large model does not **fit into memory → Pipeline and Tensor parallelism** needed
- Tensor parallelism: More communication → Distribution within node
- Pipeline parallelism: Less arithmetic efficiency ("Pipeline bubble").
- Challenge: A100 only 40GB



Strong scaling for 13B model

Model layout for 13B parameters on 8 nodes, 32 GPUs

- Parameters + gradients + optimizer states = 56 GB (ZeRO Stage 1)
- MP = 4 reduces it to 14 GB per GPU.



Challenges and Collaborations

Sequana 2 cabinet has a **hardware problem** with flipping links.
- Spurious error showing up as port error in NCCL
- Hard to reproduce, even harder to "debug"
- **Reproducing** SOTA vs. **novel research**
- **Energy** consumption
  - GPT-3 training = power for > 100 houses for a year.

```
jwb0694:13939:14102 [0] transport/net_ib.cc:94 NCCL WARN NET/IB : Got async event : port error
jwb0694:13938:14101 [0] transport/net_ib.cc:94 NCCL WARN NET/IB : Got async event : port error
jwb0694:13940:14103 [0] transport/net_ib.cc:94 NCCL WARN NET/IB : Got async event : port error
jwb0694:13941:14108 [0] transport/net_ib.cc:94 NCCL WARN NET/IB : Got async event : port error
```

From Wikimedia Commons, Rahm Emanuael

Possible collaborations: Do you have experience with LLMs or interesting ML-specific hardware?