Topics for AI/ML on HPC systems

Grigory Shamov,
October 23, 2026





University of Manitoba


Artificial Intelligence and Machine Learning revolution
https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/35179.pdf "The Unreasonable Efficiency of Data"
http://www.incompleteideas.net/IncIdeas/BitterLesson.html "The Bitter Lesson"
https://dl.acm.org/doi/10.5555/3295222.3295349 "Attention is all you need"

Infrastructure for AI is the same HPC with GPUs

***tamIA**, an AI Supercomputer of LavalU*

AI is an HPC workload :

- Clusters, managed either by SLURM or Kubernetes. Scale-out parallel storage filesystems. Fast interconnects
- *“All of it we had in HPC! We did the right thing! “*



This made old-guard HPC folks happy. Back in 2024.

What changed since?

AI workloads meant model Training and Inference

- Training shops did ok in HPC settings (either SLURM or Kubernetes)
- Early 2026, a large shift to inference!
- RAGs, Vibe Coding using public Inference services/APIs
- Agentic AI, coding and automation

The "Sovereign AI" idea, to decrease the risks of using US Big Tech

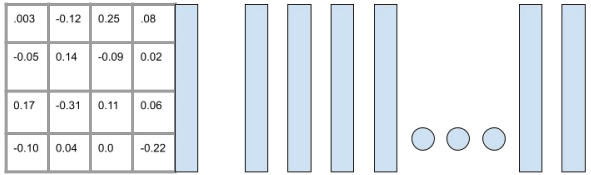
- Canada has issued 80M stop-gap funding in 2025
- AI SCIP program call, 800+M$ is open now
    - https://ised-isde.canada.ca/site/ised/en/ai-sovereign-compute-infrastructure-program

Inference is profitable: caused AI bubble with hardware costs! Canada has issued 80M stop-gap funding in 2025

- Hyperscalers ate all the hardware, exceed size of Top500 systems
- Getting electricity and cooling for data centers became an issue!

What are those AI and “models”

- Layers of numerical weights, and metadata about them
- Specific to the model’s software and hardware (precision)
  - (“transformers” would have attention layer, etc.)



- **Training/tuning** optimizes the weights of all or some layers
- **Inference** converts a prompt into a numerical representation , passes through model layers and generates outputs

https://poloclub.github.io/transformer-explainer/

What are those AI and “models”

- **Training/tuning** optimizes the weights of all or some layers

- **LoRA, QLoRA** (Low Rank Adaptation of Large Language Models)
  - optimizes weights of a few layers to improve an existing model

Training on HPC is a natural fit

- Needs fast GPUs, needs fast storage and interconnect.
- Can be done with **SLURM**, Batch Mode, as a job, using many GPUs
- Python software (pytorch, diffusers, transformers, accelerate, sloth) - vurtualenv

- If the training is really what you need….

What are those AI and “models”

- **Inference** converts a prompt into a numerical representation , passes through models layers and generates outputs

- **RAG** (Retrieval Augmented Generation) augments the prompt context with an external knowledge (numerically encoded into a “vector database” by another model)
  - Provides more context. Success depends on the “context window” of a given model.

- Inference workloads tend to be a **Service**, a Web server or API
  - Grok, ChatGPT, or Claude in VSCode
  - Local model servers like **llama.cpp**, **vLLM**, ollama, etc.
  - Local RAG, search etc.: **OpenWebUI**, **Unsloth Desktop,** etc,

Agentic AI

- Agentic AI more or less “builds” on RAG, creating an **agentic loop** using a **“harness”**
  - Reasons, calls tools , updates the context (memory) iteratively
  - Can even self-evolve skills, or self-optimize the model

- Can execute tools: if you can do a CLI work, agent can do it too.
  - Or can use MCP protocol with tools that support it

- Can be multi-agentic with “teams” of agents
  - PM managing codes, QA and benchmarking

HPC use cases?

- Internal use: analysis of user's usage;
  - A predictive, dynamic job Scheduling
  - RAGs for HPC documentation.
- Training large models; LoRAs
- Developing parallel codes for HPC.
  - Stating intent rather than low-level programming
  - From prompt/copilot to automatic agentic generation and testing
    - An example : https://github.com/Katagiri-Hoshino-Lab/VibeCodeHPC
- Automating usage of HPC systems
  - (https://clusterpilot.sh/)
  - Tools like Globus-MCP for agents https://github.com/globus/globus-mcp
- AI co-Scientists on HPC
  - Automating running computational experiments on HPC (i.e. LAMMPS)
    - https://sc25.supercomputing.org/proceedings/posters/poster_files/post284s2-file3.pdf

.

Issues with AI / ML agentic uses on HPC

- Policy and security issues
  - Using commercial models leaks data (esp for agentic use)
  - Responsibility wrt ToS ? Agents runs as a user. Safety / sandboxing
  - Network tunnels, (lack of) endpoint control, again DLP
- Ethical and organizational rules (check your Organization's AI guidelines!)
- Technical issues
  - Running microservices is harder in batch jobs
  - Local models require a lot of GPU resources to be useful
- Positives.
  - There are free GPUs, storage and networks in HPC.
  - Agents consume CPU and memory but LLMs run elsewhere, ok.
  - LLMaaS coming about: https://inference.vulcan.alliancecan.ca/

Hands-on: a local LLM with llama.cpp and Pi harness

- Download the models in a proper GGUF format from HuggingFace

- Run a SLURM job that starts **llama.cpp** service on the model
- Connect to the llama.cpp vand chat with it.

- Install and setup agentic harness, **PI**
- Connect PI to the model delivered by llama.cpp
    - Experiment with Skills and Prompts


University
of Manitoba