HPC Experiences with Intel GPU Max for Deep Learning at Scale

Communication, I/O, and Storage at Scale on Next-Generation Platforms
Scalable Infrastructures, ISC'24

*Nicholas Charron, **Steffen Christgau***

National High-Performance Computing Center @ Zuse Institute Berlin

GPUs at NHR@ZIB

- Two GPU partitions as part of *Lise* system
  - Since June 2023: 42 nodes of four Nvidia A100
  - Since March 2024: 8 nodes of four Intel PVCs → first publicly available PVC installation in DE
  - InfiniBand HDR-200 fabric
  - Host systems differ in CPU/RAM
- Small yet vendor-diverse installation → encourage users to use vendor-neutral solutions.
- More and more users involved with AI/ML

- This talk: study of AI/ML use-cases at scale: experiences + comparison with A100
- Yesterday: PVC user group talk: usability of AI/ML frameworks on PVC

Content of Study

- Usability: Do PyTorch frameworks work with Intel GPU Max as well?
- How does performance compare with Nvidia A100?
- Three use cases:
    1. Training: RESNET-50 (Conv. DNN) with CIFAR10 input set: fixed batch/image size classification, 512 for A100 and 2x 256 for 2-tile PVC
    2. Inference (with BF16) for network above
    3. Training: SAGE (GraphNN): variable batch/input size classification, 2048 subgraphs for A100 and 2x 1024 for 2-tile PVC

- Use cases scaled up to 8 (nodes) × 4 (GPUs per node) = 32 GPUs (max. of PVC partition)

- Intentionally not done: deep optimization dive (take user perspective - what the typical user sees when initially migrating their code)

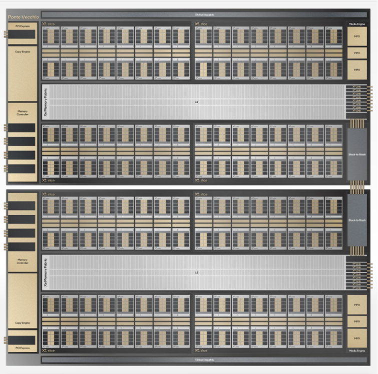
Getting it to run

- Employed framework: PyTorch
  - Minimal differences in code (`"cuda:0"` → `"xpu:0"` + additional imports)
  - See PVC user group talk for details
- Important: Intel oneCCL extension for PyTorch
- Issues with earlier oneAPI oneCLL releases; should be fixed by now in v2021.12
- Overall: **minor changes required**, but versions play a role

- Software in use:
  - Rocky Linux 8.9
  - OneAPI 2024.0.0 (with oneCCL 2021.12)
  - Intel MPI 2021.11 for both Intel and Nvidia experiments

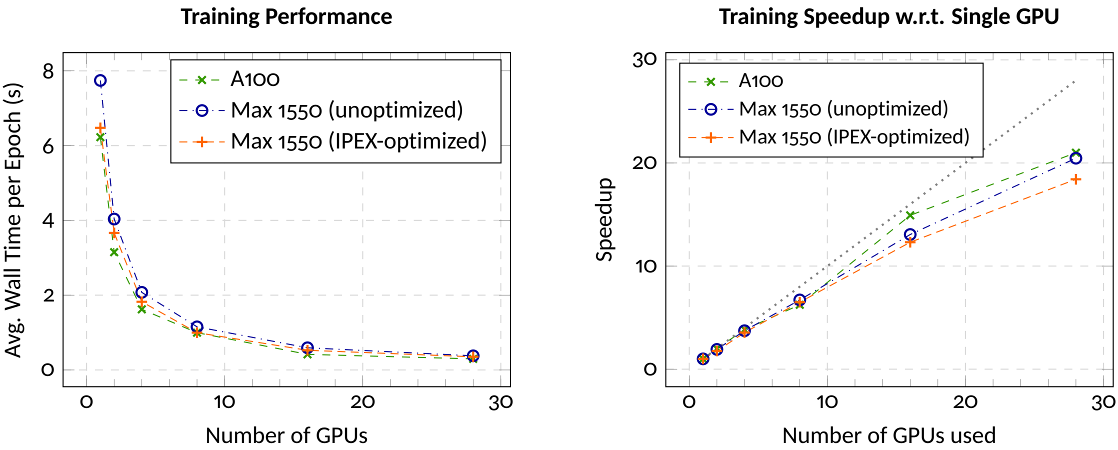
Architectural Specialties of Intel GPU Max 1550

- Max 1550 composed of two stacks (aka tiles)
- Software can have different perspectives
  - "flat": two stacks exposed individually
  - "composite": single card (+ subdevices) exposed; matches hardware architecture
  - controlled by `ZE_FLAT_DEVICE_HIERARCHY` envvar
- MPI settings should match
  - `I_MPI_OFFLOAD=1` (enable offload support)
  - `I_MPI_OFFLOAD_CELL={tile,device}`
  - check correct pinning with
    - `I_MPI_DEBUG=3`
    - `I_MPI_OFFLOAD_PRINT_TOPOLOGY=1`
- No significant difference between flat and composite



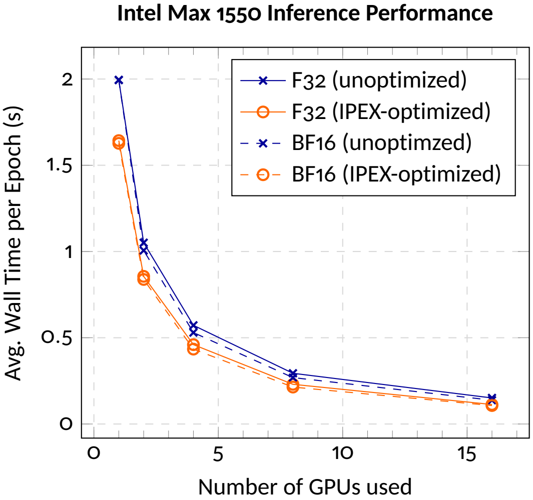
Results: Training of ResNet-50 (CIFAR-10, F32)

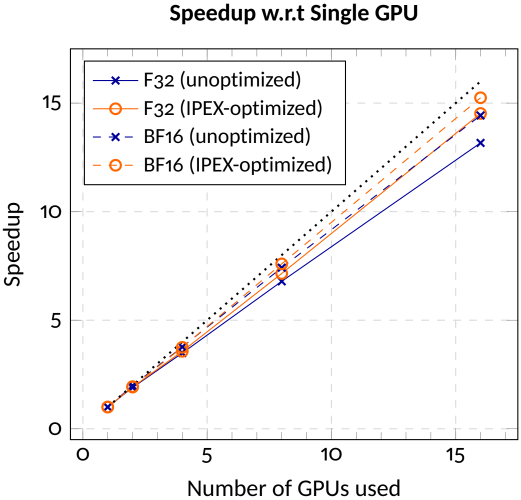
Performance and Scalability



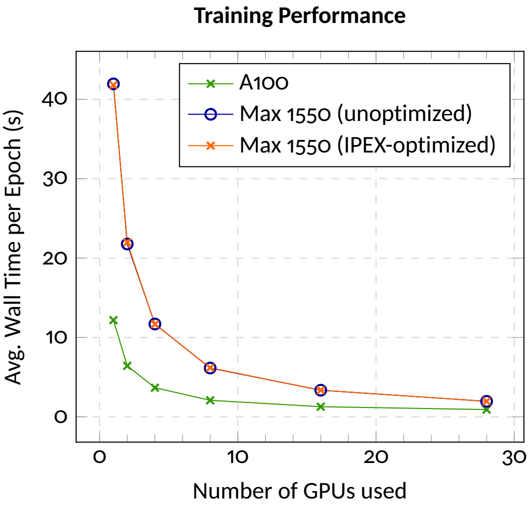
#GPUs	1	2	4	8	16	28	gmean
ipex/a100	1.04	1.16	1.12	1.0	1.26	1.19	1.12

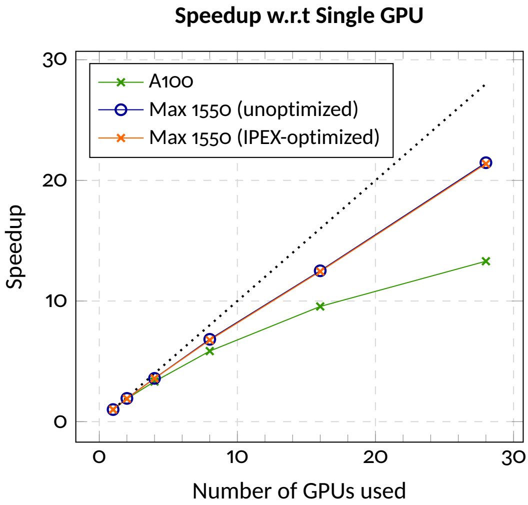
Results: ResNet-50 Inference





Results: Training of SAGE (Reddit)





Results: Training of SAGE (Reddit)

pytorch_sparse/setup.py

```
WITH_CUDA = False
if torch.cuda.is_available():
    WITH_CUDA = CUDA_HOME is not None or torch.version.hip
suffices = [’cpu’, ’cuda’] if WITH_CUDA else [’cpu’]
```

- IPEX optimizations focus mostly on image/LLM tasks, but there are not many for special graph network architectures.
- Work in progress - where are the bottlenecks?
- XPU support is not yet universal

Summary

- Max 1550s offer comparable performance + scaling to A100s for distributed AI/ML
- Max 1550s can be used for a variety of AI/ML tasks, not just LLM/Image-based inference.
- Intel extensions offer easy and tuneable options for optimizing both inference and training performances.
- Software support for XPU for certain popular pytorch libraries is still missing, but things get upstreamed

Questions? Discussion!

Contact: charron@zib.de Github: github.com/nec4/pvc_a100_comp