WW34

Keping:

2 profiling issues.

(1) Pytorch profiler + Ray Train can't work together <https://github.com/ray-project/ray/issues/47131>.

(2) HPU profiler doesn't work in Ray

Our current throughput is 10 - 24 img/sec on single card.

Stable Diffusion Habana Dashboard performance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stable Diffusion | 64 | bf16 | 11145.8 img/sec | 32 | Lightning 2.2.4 |
| Stable Diffusion Fine Tuning\* | 1 | bf16 | 71 img/sec | 7 | Lightning 2.2.4 |
| Stable Diffusion Fine Tuning Textual Inversion\* | 1 | bf16 | 20.9 img/sec | 7 | Lightning 2.2.4 |

<https://developer.habana.ai/get-started/habana-models-performance/>

LLM on Ray: Easy data PR reviewed.

Ray document update.

vLLM on Ray PR: ping anyscale to merge.

Gangsheng:

Verified llama3 can work.

Yizhong:

GPU env is ready.

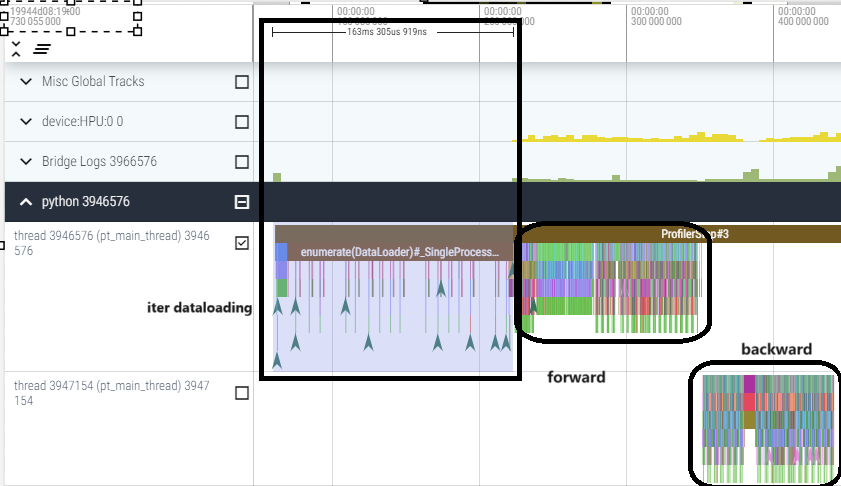
Next: run llm on ray serving and finetuning on PVC.

WW33

Keping:

I'm doing experiments of ray on gaudi. Currently I have three scripts:

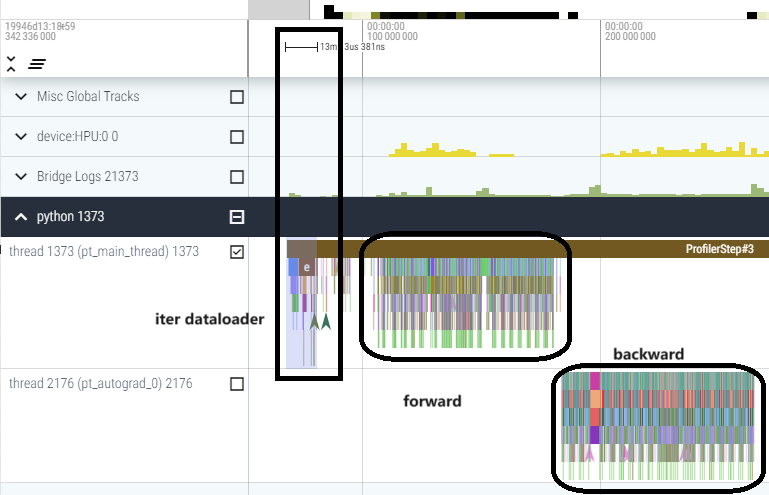
Script 1:  stable diffusion training script in optimum habana project:  <https://github.com/huggingface/optimum-habana/blob/main/examples/stable-diffusion/training/textual_inversion.py>



Script 2: Preprocess the dataset including text and image encoding operations, and save the results as parquet file.

Based on Script 1, modify the dataset to read directly from the parquet file.

This result can be used as a baseline because the HPU resources are mainly used for UNET training.



Script 3: Use ray to do end-to-end training. The data preprocessing including text and image encoding is assigned to one HPU card, and unet training is scheduled on another hpu card.

Gangsheng: Run and verify llama 3 using QA scripts this week.

Yizhong:

Follow following document to get GPU node and run Ray.

<https://wiki.ith.intel.com/display/HPDA/Borealis+Cluster>

<https://wiki.ith.intel.com/display/HPDA/Running+llm-on-ray+finetune+workflow+on+PVCs>

WW30

PyTorch Gaudi: 5 areas to work. Wil talk with pyTorch bridge managers for details.

llama3.1 70b reproduced OOM.

Ray:

Ray data: tested ray blog data processing on cpu, hpu. Will compare Ray data vs torch data

vLLM on Ray in llm-on-ray ready to review and merge.

CI: fixed CI state. Adding finetuning workflow.

Jiafu: NS+ llm on ray: identified optimization opportunity in NS for larger batch size.. Next: merge to one branch and summarize document.

1.17 vllm perf: we got 3100 tokens/s , Yaoqing: 4800 tokens/s without http server.

Aurora: tested 256 nodes kmeans but failed with Spark event issues.

DAOS: still can't access in the compute node.

`

WW29

PyTorch Gaudi project, Ramp up Gaudi stack in 1 week.

The proposal "Turbocharging GenAI workloads with Ray on Intel Gaudi accelerators" was accepted by Ray Summit.

Submit self-assessment.

Status Update:

|  |  |  |
| --- | --- | --- |
| Pretrain/Finetune: Support FP8 training | Gangsheng | Still waiting for Habana's next release (INC support) |
| Pretrain: Refactor llm-on-ray to single script, add metrics | Gangsheng | Not started |
| Finetune: Support DPO and get expected evaluation result | Minming |  |
| Finetune: Support output Throughput metrics, refresh performance with latest | Gangsheng | PR ready <https://github.com/intel/llm-on-ray/pull/265>  Next: pass CI and merge. |
| Inference: Test llama2 7b, 70b with latest optimum-habana 1.6 | Gangsheng | Issues in torch compile mode. Submitted JIRA.  Next: reproduce the issue without Ray. Check logs.. |
| Ramp up PyTorch Gaudi | Gangsheng, Xiaochang | Npu stack works. WIP Setup Qnpu. |
| OPEA Finetuning Service | Xiaochang | Initial PRs, should update to align with finetuning RFC  [opea-project/GenAIInfra#122](https://github.com/opea-project/GenAIInfra/pull/122)  [opea-project/GenAIComps#236](https://github.com/opea-project/GenAIComps/pull/236)  [opea-project/GenAIExamples#315](https://github.com/opea-project/GenAIExamples/pull/315)    Updated OPEA Finetuning RFC document.  Merged PR to add KubeRay support in OPEA. [#163](https://github.com/opea-project/GenAIInfra/pull/163) |
| vLLM: Upstream Gaudi/CPU examples to Ray | Keping | [[HPU] [Serve] Add vllm HPU support in vllm example](https://github.com/ray-project/ray/pull/45893)  1 approve, address comments.  Submit a bug about vllm on hpu in wiki (<https://jira.habana-labs.com/browse/SW-191226>) |
| vLLM Gaudi/cpu + Ray benchmark | Keping | llama2 7b  Input Token Throughput: 562.644 tokens/s  output Token Throughput: 538.132 tokens/s |
| Update vllm on Ray implementation in llm\_on\_ray | Keping | <https://github.com/intel/llm-on-ray/pull/262>  Issue in CI. Upgraded vLLM 0.5.1 to configure rc to control avx512. Upgraded to IPEX2.3 |
| Ray Data + SD | Keping | Tested Habana SD example using torch data loader. 17.5 min  Tested Ray torch trainer SD example. 19 min |
| vLLM + NS | Jiafu | * + 1. Submitted new PR with some new NS optimization merged and some review comments addressed. <https://github.com/intel/llm-on-ray/pull/267>     2. Benchmarked IDC data with eos in openai mode in NS quant and plain vllm/cpu. Same perf gain about 2x.     3. Improved deployment by adding 'inference\_engine' and 'app\_router' resources to ray actor options.     4. Fixed some issue, such as incorrect latency in open-ai mode in the benchmark script. |
| Gaudi CI | Yizhong | Inference test is working with new CI. |
| CI: Migrate self-hosted runner | Carson/Yizhong |  |
| Refactor ci and tests | Tianchen | Finish PRS,  Test Openvino vs. NS? |
| Resnet: QA issue on Gaudi |  | No update |
| Upstream Vit Gaudi example <https://github.com/ray-project/ray/pull/45381> |  | No review. |
| GNR Launch support for Ray |  | Pending GNR access. |
| Aurora | Minming/Jiafu | OPA Mllib 128 nodes can run but perf is worse than oneDAL.    DAOS: working on the verification |

WW28

Status Update:

|  |  |  |
| --- | --- | --- |
| Pretrain: Upstream llama2 example <https://github.com/ray-project/ray/pull/45459> | Gangsheng | Merged. |
| Pretrain/Finetune: Support FP8 training | Gangsheng | Still waiting for Habana's next release (INC support) |
| Pretrain: Refactor llm-on-ray to single script, add metrics | Gangsheng | Not started |
| Finetune: Support DPO and get expected evaluation result | Minming |  |
| Finetune: Support output Throughput metrics, refresh performance with latest | Gangsheng | PR ready <https://github.com/intel/llm-on-ray/pull/265>  Next: Update slides with perf data and image  Llama2 7b/70b |
| Finetune: Torch compile with Gaudi | Gangsheng | Tested with QA scripts and llm-on-ray. Results aligned.  Next: Update results in JIRA. |
| Inference: Test llama2 7b, 70b with latest optimum-habana 1.6 | Gangsheng | WIP. |
| OPEA Finetuning Service | Xiaochang | Initial PRs:  [opea-project/GenAIInfra#122](https://github.com/opea-project/GenAIInfra/pull/122)  [opea-project/GenAIComps#236](https://github.com/opea-project/GenAIComps/pull/236)  [opea-project/GenAIExamples#315](https://github.com/opea-project/GenAIExamples/pull/315)    WIP Updating OPEA Finetuning RFC  Submitted PR to add KubeRay support in OPEA. |
| vLLM: Upstream Gaudi/CPU examples to Ray | Keping | [[HPU] [Serve] Add vllm HPU support in vllm example](https://github.com/ray-project/ray/pull/45893)  1 approve, address comments.  Submit a bug about vllm on hpu in wiki (<https://jira.habana-labs.com/browse/SW-191226>) |
| vLLM Gaudi/cpu + Ray benchmark | Keping | llama2 7b  Input Token Throughput: 562.644 tokens/s  output Token Throughput: 538.132 tokens/s |
| Update vllm on Ray implementation in llm\_on\_ray | Keping | <https://github.com/intel/llm-on-ray/pull/262>  Need to fix: vllm engine cannot be serialized. |
| Ray Data + SD | Keping | Analyzing… data process: use CPU? Integrate Habana data loader? |
| vLLM + NS | Jiafu | * + 1. PR reviewing in progress. <https://github.com/intel/llm-on-ray/pull/264>     2. Fixed llama-3 performance issue in ns by extending prompt length limit from 6144 to 8192.     3. Completed IDC data test on llama 3 with vllm cpu and vllm cpu NS extension. Got 2x perf gain. [LLM on Ray (vllm cpu/vllm cpu NS extension) IDC Perf Test on llama 3](onenote:Performance.one#LLM%20on%20Ray%20(vllm%20cpu\vllm%20cpu%20NS%20extension&section-id={D09FBE29-B0DE-4E09-BBB3-8333BBC71AED}&page-id={A71B3983-BE53-4D2C-9C46-7755ED399A28}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML)     4. Chunked prefill worked well in llama.cpp, but not neural-speed. Will check in details. |
| Gaudi CI | Yizhong | * + 1. Comments of all the 3 PRs are fixed, ready to review or merge.     2. Tring Mixtral-8x7B, monitoring memory usage, tried all serve configurations, finding more memory related configs of Ray and model. |
| CI: Migrate self-hosted runner | Carson | Implemented scripts to pull PRs, run CI tests and post status. |
| Refactor ci and tests | Tianchen | Fix PR219 comments. Test debug\_mode bugs |
| Resnet: QA issue on Gaudi |  | No update |
| Upstream Vit Gaudi example <https://github.com/ray-project/ray/pull/45381> |  | No review. |
| GNR Launch support for Ray |  | Pending GNR access. |
| Aurora | Minming/Jiafu | 1-64 nodes tests kmeans ready, aligned oneDAL result  Plan: testing other 3 algos. 128 nodes jobs pending on resource.  DAOS: still unavailable. |

WW27

Status Update:

|  |  |  |
| --- | --- | --- |
| Pretrain: Upstream llama2 example <https://github.com/ray-project/ray/pull/45459> | Gangsheng | Pending to merge, all comments addressed. |
| Pretrain/Finetune: Support FP8 training | Gangsheng | Still waiting for Habana's next release (INC support) |
| Pretrain: Refactor llm-on-ray to single script, add metrics | Gangsheng | Not started |
| Finetune: Improve evaluation result for SFT | Minming, Gangsheng | Merged PR <https://github.com/intel/llm-on-ray/pull/252> |
| Finetune: Support DPO and get expected evaluation result | Minming | Locally tested. Still gap on the result. |
| Finetune: Support output Throughput metrics, refresh performance with latest | Gangsheng | Submitted PR <https://github.com/intel/llm-on-ray/pull/258>  Llm-on-ray Llama2 7b: fixed  Llm-on-ray Llama2 70b: fixed.  Bugs fix PR: <https://github.com/intel/llm-on-ray/pull/265> |
| Finetune: Torch compile with Gaudi | Gangsheng | Not started |
| Inference: Test llama2 7b, 70b with latest optimum-habana 1.6 | Gangsheng | Not started |
| OPEA Finetuning Service | Xiaochang | Submitted initial PRs:  [opea-project/GenAIInfra#122](https://github.com/opea-project/GenAIInfra/pull/122)  [opea-project/GenAIComps#236](https://github.com/opea-project/GenAIComps/pull/236)  [opea-project/GenAIExamples#315](https://github.com/opea-project/GenAIExamples/pull/315)    Synced with Tian, Feng on writing OPEA Finetuning RFC |
| vLLM: Upstream Gaudi/CPU examples to Ray | Keping | [[HPU] [Serve] Add vllm HPU support in vllm example](https://github.com/ray-project/ray/pull/45893)  1 approve, address comments.  Submit a bug about vllm on hpu in wiki (<https://jira.habana-labs.com/browse/SW-191226>) |
| vLLM Gaudi/cpu + Ray benchmark | Keping | Gaudi: WIP: run benchmark using vllm-fork master.  CPU: iomp5.so, 1 replica 53core: 269.037 tokens/s-> 291.986 tokens/s, 2 replicas 1 socket: no improvement. |
| Update vllm on Ray implementation in llm\_on\_ray | Keping | <https://github.com/intel/llm-on-ray/pull/262>  Need to fix: vllm engine cannot be serialized. |
| vLLM + NS | Jiafu | * + 1. PR ready for review. <https://github.com/intel/llm-on-ray/pull/264>     2. Supported llama3-8b-instruct. Verified all IDC test data.     3. Tuning code for long prompt, including chunked prefill and dynamic context size -> page attention.     4. Chunked prefill worked well in llama.cpp, but not neural-speed. Will check in details. |
| Gaudi CI | Yizhong |  |
| CI: Migrate self-hosted runner |  |  |
| Refactor ci and tests | Tianchen | Fixing PR comments. |
| Resnet: QA issue on Gaudi |  | No update |
| Upstream Vit Gaudi example <https://github.com/ray-project/ray/pull/45381> |  | No review. |
| GNR Launch support for Ray |  | Pending GNR access. |
| RayDP: | Carson | Released 1.6.1 |
| Aurora | Minming/Jiafu | 1-64 nodes tests kmeans ready, aligned oneDAL result  Plan: testing other 3 algos. 128 nodes jobs pending on resource.  DAOS: still unavailable. |

WW26

• Passdown& open

○ AIQT passdown

§ Unified SW architecture - still at a point of collecting information

§ Zane ball open forum

§ ESS results - not good

§ June - project review for all new projects

○ DCAI passdown

§ 2nd half year priority

§ Xeon6

§ Gaudi2 deal

§ Maintain MSS for key customer

§ ACS intro

§ Combine with "DCAI China Grow24|AI大咖说 “Gaudi智算未来” to learn more

Status Update:

|  |  |  |
| --- | --- | --- |
| Pretrain: Upstream llama2 example <https://github.com/ray-project/ray/pull/45459> | Gangsheng | Another reviewer left comments to follow up. |
| Pretrain/Finetune: Support FP8 training | Gangsheng | Will wait for Habana's next release (INC support) |
| Pretrain: Refactor llm-on-ray to single script, add metrics | Gangsheng | Not started |
| Finetune: Improve evaluation result for SFT | Minming, Gangsheng | Submitted PR <https://github.com/intel/llm-on-ray/pull/252>  Ready to merge. |
| Finetune: Support DPO and get expected evaluation result | Minming |  |
| Finetune: Support output Throughput metrics, refresh performance with latest | Gangsheng | Submitted PR <https://github.com/intel/llm-on-ray/pull/258>  Llm-on-ray Llama2 7b: perf aligns, but problem in eval.  Llm-on-ray Llama2 70b: failed. |
| Finetune: Torch compile with Gaudi | Gangsheng | Not started |
| Inference: Test llama2 7b, 70b with latest 1.6 | Gangsheng | Not started |
| OPEA Finetuning Service | Xiaochang | Submitted initial PRs:  [opea-project/GenAIInfra#122](https://github.com/opea-project/GenAIInfra/pull/122)  [opea-project/GenAIComps#236](https://github.com/opea-project/GenAIComps/pull/236)  [opea-project/GenAIExamples#315](https://github.com/opea-project/GenAIExamples/pull/315)    Synced with Tian, Feng |
| vLLM: Upstream Gaudi/CPU examples to Ray | Keping | [[HPU] [Serve] Add vllm HPU support in vllm example](https://github.com/ray-project/ray/pull/45893) |
| vLLM Gaudi/cpu + Ray benchmark | Keping |  |
| Update vllm on Ray implementation in llm\_on\_ray | Keping | PR ready: <https://github.com/intel/llm-on-ray/pull/262> |
| vLLM + NS | Jiafu | * + 1. Merged inference engine with ns kernel optimization     2. Benchmarked llm-on-ray/vllm/ns before code merge and after code merge.     3. Added feature of quantizing model on the fly with multiple instances.     4. Preparing PR to merge |
| Gaudi CI | Yizhong |  |
| CI: Migrate self-hosted runner |  |  |
| Refactor ci and tests | Tianchen | Help fix oneapi bugs. PR219 not merged because this bug, Delete HF\_token output |
| Resnet: QA issue on Gaudi |  |  |
| Upstream Vit Gaudi example <https://github.com/ray-project/ray/pull/45381> |  | No review. |
| GNR Launch support for Ray |  | Pending GNR access. |
| RayDP: | Carson | Released 1.6.1 |
| Aurora | Minming/Jiafu | Env is ready. Scripts updated. Start to run 128 node scale.  DAOS: contact daos team… |
|  |  |  |

WW25:

Encouraged Wellness Day: 7/8, 9/2

Status Update:

|  |  |  |
| --- | --- | --- |
| Pretrain: Upstream llama2 example <https://github.com/ray-project/ray/pull/45459> | Gangsheng | No Update |
| Pretrain/Finetune: Support FP8 training | Gangsheng | Tested Habana example: 10% improvement in finetuning  AR: sync with Jerome. |
| Pretrain: Refactor llm-on-ray to single script, add metrics | Gangsheng | Not started |
| Finetune: Improve evaluation result for SFT | Minming, Gangsheng | Submitted PR <https://github.com/intel/llm-on-ray/pull/252>  Result aligned with Neural chat. reviewing |
| Finetune: Support DPO and get expected evaluation result | Minming | Ran the ray version. Checking result. |
| Finetune: Support output Throughput metrics: Samples/s | Gangsheng | Submitted PR <https://github.com/intel/llm-on-ray/pull/258> |
| Finetune: Habana QA Test | Habana QA, Gangsheng | QA team started to test. Updated scripts based on QA feedback. |
| OPEA Finetuning Service | Xiaochang | Updated demo slides, video. |
| vLLM: Upstream Gaudi/CPU examples to Ray | Keping | [[HPU] [Serve] Add vllm HPU support in vllm example](https://github.com/ray-project/ray/pull/45893) No review. |
| vLLM Gaudi/cpu + Ray benchmark | Keping | CPU: tested tcmalloc, perf improved.throuhput on SPR : 576 -> 663 tokens/s  LORA adapter works on CPU. |
| Update vllm on Ray implementation in llm\_on\_ray | Keping | Not started. |
| vLLM + NS | Jiafu | * + 1. Tried snc4 and snc2, no apparent improvement     2. Remove unnecessary OMP thread sync for kernel init and finalize. Got about overall 6% improvement with 8 batch-size vllm offline inference.     3. Identified scalability issue of FFN\_SILU kernel. Tried latest neural speed code, but perf got dropped.     Next: generate a report for current status. |
| Gaudi CI | Yizhong |  |
| Refactor ci and tests | Tianchen | 2PRs fix some bugs merged, [PR244](https://github.com/intel/llm-on-ray/pull/244),[PR221](https://github.com/intel/llm-on-ray/pull/221), one will be merged today[PR219](https://github.com/intel/llm-on-ray/pull/219) (about docker\_user need check hf\_token) |
| Resnet: QA issue on Gaudi |  |  |
| Upstream Vit Gaudi example <https://github.com/ray-project/ray/pull/45381> |  | No review. |
| GNR Launch support for Ray |  | Pending GNR access. |
| Fix detected secret in build log - intel/llm-on-ray | Jiafu | Done |
| Llm-on-ray SDLe Refresh | Carson | Done |
| RayDP: | Carson | Spark 3.5 support PR merged. Plan to release 1.6.1 |
| Aurora | Minming | Still can't access lustre queue. |

WW24:

Status Update:

|  |  |  |
| --- | --- | --- |
| Pretrain: Upstream llama2 example <https://github.com/ray-project/ray/pull/45459> | Gangsheng | Pending on review. |
| Pretrain/Finetune: Support FP8 training | Gangsheng | Testing, updated habana driver, issue on memory allocation  No official guide for FP8, need experiment. |
| Pretrain: Refactor llm-on-ray to single script | Gangsheng | Not started |
| Finetune: Improve evaluation result for SFT | Minming, Gangsheng | Root cause: no padding in our code and need to fill attention mask |
| Finetune: Support DPO and get expected evaluation result | Minming | PR submitted. Fixing CI. |
| Finetune: Support output Throughput metrics: Samples/s | Gangsheng | Several metrics for throughput, sentence/token per sec.    sentences/sec: total number of samples divided by total training time  tokens/sec: total number of tokens divided by total training time |
| OPEA Finetuning Service | Xiaochang | Demo video done. |
| vLLM: Upstream Gaudi/CPU examples to Ray | Keping | [[HPU] [Serve] Add vllm HPU support in vllm example](https://github.com/ray-project/ray/pull/45893) |
| vLLM Gaudi/cpu + Ray benchmark | Keping | No update |
| vLLM + NS | Jiafu | Improved perf (latency and throughput) by reducing remote numa access.    32 input/64 output (with ray vllm example and openai call):  8 batch: 32 users/75ms/375 tokens  4 batch: 16 users/54ms/238 tokens  1024 input/128 output (with llm-on-ray):  4 batch: 16 users/76ms/162 tokens |
| Gaudi CI | Yizhong | 2 issues. Mixtral and Gaudi Predictor model output |
| Refactor ci and tests | Tianchen | 2PRs ready, [PR244](https://github.com/intel/llm-on-ray/pull/244),[PR221](https://github.com/intel/llm-on-ray/pull/221), one will ready today [PR243](https://github.com/intel/llm-on-ray/pull/243),[PR219](https://github.com/intel/llm-on-ray/pull/219) need more review |
| Resnet: QA issue on Gaudi |  | Sync with Linzhi |
| Upstream Vit Gaudi example <https://github.com/ray-project/ray/pull/45381> |  | Sync with Linzhi |
| SRF, GNR Launch support for Ray |  |  |

WW23

Pretraining:

- PR: removed megatron, fixed output.

- llm-on-ray refactor: WIP

Finetuning:

- Improving evaluation result: habana example is good.

AR: Minming, Gangsheng: send out a plan for next week WW23.5 ([Plan](onenote:#LLM-on-Ray%20fine-tuning%20evaluation&section-id={08438F69-9374-47E5-83E4-DE309A4A75B0}&page-id={EB68E5F8-7C27-42C5-BA94-E56A95C6760C}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML/Plan.Status.Update.one))

* + DPO: resolved comments.
  + OPEA Service:
    1. Demo: the pipeline can run successfully. To discuss how to record and show value.

Serving:

* + vLLM: upgraded habana 1.15 to 1.16. vllm fork, ray example + vllm, llm-on-ray + vllm have the same perf now: llama2 base: 493 tokens/s.
    1. Open: llm-on-ray: Switch to ray upstream example?
    2. Next: test TP with Gaudi. Upstream: add Gaudi support in the example.
  + vLLM + NS: Got initial benchmark result

Gaudi:

* + Resnet QA issue: lazy, eager mode can work. Compile still can't work.

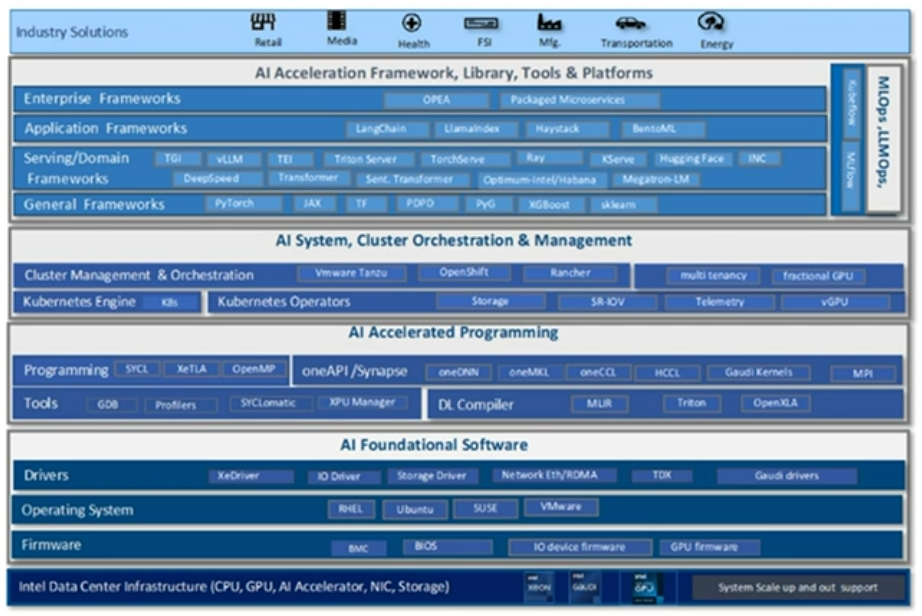
CI:

* + Inference: PR is ready to review. Debugging issues in two models.

Open: check model output?

Tiancheng: 4 PRs, being reviewed.

WW22



LLM on Ray Finetuning

* + OPEA Finetuning Service Integration
    1. POC: build finetune microservice with fastapi+raytrain
       - OpenAI finetuning protocol: create job (done)
       - Query job status: callback works, also support tensorboard/wandb to check finetuning process (done)
       - KubeRay autoscaling (done)
       - Slides (wip)
  + Refactor, merged
  + Submit
  + Evaluation, still running

Finetuning & evaluation:

Next Steps

* + Check data in each batch. Check parameters passed to GaudiTrainer. (Gangsheng)
  + Run Habana finetuning script. (Minming)
  + DPO: PR submitted. need rebase.

Removed deltatuner. PR merged.

Pretraining

* + Refactoring LLM-on-Ray workflow, become single script. (2 code pathes)
  + Upstreaming: pending review. (megatron-deepspeed vs. Habana's version ?)
  + Update Performance

Gaudi:

* + Llama
    1. Pre-training [PR](https://github.com/ray-project/ray/pull/45459/files), reviewing
    2. Fine-tuning [PR](https://github.com/ray-project/ray/pull/44667), merged
  + SD merged.
  + ViT

VLLM: tested habana\_main performance, worse than expected. WIP tuning.

VLLM + NS: Identified root cause of slowness in NS for traversing nodes. Debugging and optimizing.

CI: recovered CI

Inference: Mistral 7b OOM on single card , Qianwen 110b failed with HTTP 500 error.

Also need to check result correctness.

Finetune: WIP implementing the workflow.

WW21

* + Take the Employee Experience and Employee Inclusion Surveys

LLM on Ray Finetuning :

* + Refactor [PR](https://github.com/intel/llm-on-ray/pull/233)
    1. Data Collator
       - Label fields reset to –100 before response field
    2. Remove tokenizer/model module dependency in common
  + Resolve the exception caused by missing validation data and the missing gaudi\_config parameter in GaudiTrainer, submit and fix comments [PR #227](https://github.com/intel/llm-on-ray/pull/227), pending to review.
  + Completed DPO trainer coding and debug now.
  + OPEA Finetuning Service Integration
    1. Setup Ray cluster with KubeRay in K8S (DONE)
    2. Build finetune microservice with fastapi (WIP)
       - OpenAI finetuning protocol: create job, list job, query status

LLM on Ray Serving

* + Enable vllm guadi in llm-on-ray: [PR#232](https://github.com/intel/llm-on-ray/pull/232), ready.
  + Add autocaling-config: [PR#214](https://github.com/intel/llm-on-ray/pull/214), ready.
  + Benchmark end to end script [PR#195](https://github.com/intel/llm-on-ray/pull/195), move ci test to self-host machine.

LLM on Ray Quantization

* + Integrated vllm/inference-engine into llm-on-ray. PR submitted
  + Passed all test cases in the shareGPT dataset, fixed some bugs in the integration and inference engine.
  + Working on next token generation performance tuning

Gaudi:

* + Llama
    1. Pre-training [PR](https://github.com/ray-project/ray/pull/45459/files), pending to review
    2. Fine-tuning [PR](https://github.com/ray-project/ray/pull/44667), pending to merge
  + ViT
    1. Ping Cheng to take a look. CI seems to pass
  + Stable Diffusion
    1. PR ready to merge, got 1 approval
    2. Ray Data optimization: [Reducing the Cost of Pre-training Stable Diffusion by 3.7x with Anyscale](https://www.anyscale.com/blog/scalable-and-cost-efficient-stable-diffusion-pre-training-with-ray)
  + Finetuning Mistral
    1. llm-on-ray leaderboard result: [Picture on page "Open LLM Leaderboard Results"](onenote:Minming.one#Open%20LLM%20Leaderboard%20Results&section-id={5785A99B-C365-46B8-91DD-C034A272F7BB}&page-id={41FB65E6-6226-4B50-89BF-88CA5BD16A79}&object-id={6376C124-599E-0594-1D10-E82C172D74F6}&97&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML)
    2. Running Intel/neural-chat-7b-v3-1 on Intel Gaudi2, but it will print error.

文本

描述已自动生成

Tests/CI:

FIX [PR#219](https://github.com/intel/llm-on-ray/pull/219) reviews,Set up separate dockerfile for user**,[TODO]** fix keep\_serving\_run bugs about docker run with entrypoint

Add serve command line options to list all supported model-ids [PR#221](https://github.com/intel/llm-on-ray/pull/221) ，passed

CI,Change miniconda to miniforge [PR#230](https://github.com/intel/llm-on-ray/pull/230),passed

Define simple\_protocol.py [issues#217](https://github.com/intel/llm-on-ray/issues/217)

WW20

* + Pallavi Org, Sid Update
  + Rui/Peter QGS
  + Back to office plan
  + Cloud account:
    1. re-enforce to delete the instances not used.
    2. We will STOP use Google & Azure instances starting from June, and only use AWS.
  + Some Aurora Nodes become available.
  + Gaudi node 0 & 1 will be shared with Chendi's team.

LLM on Ray Pre-training:

* + Fixed some bugs

LLM on Ray Finetuning :

* + Completed SlimOrcaDataPreprocess coding with format dataset Open-Orca/SlimOrca.

LLM on Ray Serving

* + Refactor
    1. Non-streaming generate (Merged)
    2. Dockerfile consolidation support for both CI and User (WIP)
    3. TODO: streaming generate
  + Add autoscaling config([PR#214](https://github.com/intel/llm-on-ray/pull/214)): Figured out the reasons for two problems. Ready to review.
  + First verify vllm guadi in llm-on-ray: work in progress
  + Merged [PR#199](https://github.com/intel/llm-on-ray/pull/199): Integrated chat template in llm-on-ray inference.

LLM on Ray Quantization

* + Completed vllm/inference-engine integration, including offl

Gaudi:

* + Llama-2 fine-tuning:
    1. Reviewer comments:
       - Remove cell outputs, just left import outputs in mock cell
       - Merge 2 examples into 1 example, support both different training method in different configs
       - Should support not lora training?
    2. Running mistralai/Mistral-7B-v0.1 with dataset Open-Orca/SlimOrca(2.91M rows) on 7 Gaudis( 1 epoch about 2 hour)
  + Llama-2 pre-training:
    1. Create the basic scripts that can run
  + Stable Diffusion:
    1. PR submitted, about to merge; pending anyscale to solve build failure
    2. Performance updated to confluence
  + ViT:
    1. PR submitted, add instructions for HPU to the existing example
  + Inference
    1. [Inference] Remove simple mode and use OpenAI API ([PR#220](https://github.com/intel/llm-on-ray/pull/220))
    2. [Inference] Add validated models for Gaudi ([PR#225](https://github.com/intel/llm-on-ray/pull/225))

Tests:

Build docker files for both CI and User [PR#219](https://github.com/intel/llm-on-ray/pull/219), Merge the dockerfile for user (cpu\deepspeed\ipex-llm\vllm)

Add serve command line options to list all supported model-ids [PR#221](https://github.com/intel/llm-on-ray/pull/221) , List all support model\_ids with config file path:"llm\_on\_ray-serve --list-model-ids"

Define simple\_protocol.py [issues#217](https://github.com/intel/llm-on-ray/issues/217)

WW19

Back to office: Expect to be announced on July 1.

Weekly

LLM on Ray Serving:

* + RayLLM upstream
    1. PR ready for review: [#149 Update to latest vLLM upstream and Support vLLM on CPU](https://github.com/ray-project/ray-llm/pull/149)
  + vLLM integration
    1. Use latest vLLM for llm-on-ray: [Update vllm to use latest upstream to support CPU #179](https://github.com/intel/llm-on-ray/pull/179)
    2. PR to address OpenAI protocol for vLLM bug (Keping [PR#213](https://github.com/intel/llm-on-ray/pull/213))
    3. [Update VLLM installation script and documentation by xwu99 · Pull Request #212 · intel/llm-on-ray (github.com)](https://github.com/intel/llm-on-ray/pull/212)
       - Use conda to install GCC 12.3
       - Should install vLLM after installing llm-on-ray
  + Refactor
    1. [Refactor non-streaming generate by xwu99 · Pull Request #209 · intel/llm-on-ray (github.com)](https://github.com/intel/llm-on-ray/pull/209)
       - Convert str to list before generate
       - Define types for generate input and streamline preprocess\_prompts
       - Move all predictor classes to predictors directory
  + Gaudi
    1. Add support for quantization & torch\_compile, merged
    2. Found problem why 7b models not working with latest optimum habana: delete stopping criteria
  + Fixed [PR#199](https://github.com/intel/llm-on-ray/pull/199) comments and pending CI test.
  + Investigated Model Evaluation and finetuned Llama-2-7b-chat-hf on Gaudi and using [lm-evaluation-harness](https://github.com/EleutherAI/lm-evaluation-harness) to evaluate.
  + Run benchmark script: [PR#195](https://github.com/intel/llm-on-ray/pull/195), use --result\_dir to parse to benchmark result, and add CI.
  + Add autoscaling config: [PR#214](https://github.com/intel/llm-on-ray/pull/214), ray can autoscale replicas as expected. also need to validate openai api.

LLM on Ray Fine-tuning:

* + Refactor: replace fine-tuning DefaultTrainer with transformers.Trainer
    1. PR [#204](https://github.com/intel/llm-on-ray/pull/204), to be checked in CICD

LLM on Ray Quantization

* + Removed pybind11 in our inference-engine since we use ctypes instead
  + Vllm/inference-engine can generate first token successfully
  + Working on next token stuffs, like block manager, kv cache copying and slot releasing

Gaudi:

* + LLama-2 fine-tuning:
    1. PR: [#44667](https://github.com/ray-project/ray/pull/44667), added a new example of fine-tuning llama-2-70b with deepspeed on HPUs, PR will start to review this week by Ray reviewer.
  + GPT-2 pre-training:
    1. Trying to integrate pre-training GPT-2 on HPUs with Ray, and will also upstream this example to Ray community.
  + Stable Diffusion:
    1. Submitted PR:[[HPU] [Train] Add a Stable Diffusion fine-tuning and serving example by kira-lin · Pull Request #45217 · ray-project/ray (github.com)](https://github.com/ray-project/ray/pull/45217)
    2. Show how to make changes to run examples on ray; do not involve details of failing, such as failing to serialize, initializing distributed twice

Tests:

Fix some reviews, Finish docker2sh PR[123](https://github.com/intel/llm-on-ray/pull/123), Simplified workflow code

* + Config Gaudi CI node, reinstall and fix habana environment issue, add to CI node list.
  + Add llama-2-70b to inference workflow, [PR#208](https://github.com/intel/llm-on-ray/pull/208)
  + Trying to support Falcon-7B, Falcon-40B, Bloom-7B, Bloomz-176B

WW17

Potential WFH change

Mt.Sill change

OPEA

Ray Summit: 2 proposals.

Open:

OKR - workday.

Status Update:

==============

Ray upstream Gaudi:

* + Stable Diffusion:
    1. Jerome said we should educate users to adapt existing Gaudi examples to ray, e.g. optimum-habana ones
    2. Both with and w/o ray script fail in Habana VM, but it's working in our Gaudi machine
    3. Ray Serve is simpler than Ray Train
  + LLama2-70b finetuning
    1. [Updated](https://gerrit.habana-labs.com/#/c/412835/) Habana benchmark scripts for QA:
       - Support Llama-2-7b and Llama-2-70b fine-tuning with DDP and DEEPSPEED on same script.
    2. TODO: Update Ray PR to include 70b finetuning.

LLM on Ray Serving:

* + RayLLM upstream
    1. Run RayLLM docker with NVIDIA V100 successfully with GPT2, Llama2 out of memory
    2. Locally run RayLLM latest 0.5.0 backend successfully with vLLM 0.2.7 (upstream latest is 0.4.1)
       - Need to fix OMP\_NUM\_THREADS
    3. Vllm 0.4.1 support needed to upgrade RayLLM models from pydantic v1 to pydantic v2 (ww18)

* + Use vLLM upstream in llm-on-ray: TODO: Validate and merge <https://github.com/intel/llm-on-ray/pull/179>

* + Update benchmark script: [pr#195](https://github.com/intel/llm-on-ray/pull/195), more easier to use now. todo: will parse output from json/csv format.
  + Re-run the performance of multi-replicas. Need to address OMP\_PROC\_BIND problem.

* + Submitted PR [#205](https://github.com/intel/llm-on-ray/pull/205) that integrate web ui with chat template.
  + Modified PR [#199](https://github.com/intel/llm-on-ray/pull/199) comments.

* + Llama-3-8b verified to work. If we use main branch of optimum-habana, then llama-3-8b failed but llama-3-70b becomes working

* + Validation dynamic batching on guadi: [dynamic batching](onenote:Keping.one#dynamic%20batching&section-id={4C5A26D5-01E6-44D8-B726-942506655618}&page-id={15251928-29FA-4B99-8A96-87A967885D02}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML).

LLM-on-Ray Quantization:

* + Monkey patching vllm to support our own inference-engine, including ns\_loader, ns\_model and ns\_quant
  + Create our own inference-engine from neural-speed by stripping out cont. batching scheduler and sampling, python build with cmake ext passed
  + Track ns bug fixing

LLM on Ray Fine-tuning:

* + Refactor: replace fine-tuning DefaultTrainer with transformers.Trainer (optimum.habana.transformers.Trainer)
    1. PR [#204](https://github.com/intel/llm-on-ray/pull/204)
    2. Validated functionality on CPU and HPU, to be validated on GPU.
    3. Also need to refactor the fine-tuning yaml config file, can support more arguments for training
    4. Also need to refactor the finetune.py to make management of different device and training method for fine-tuning more readable.

Tests:

* + [PR109](https://github.com/intel/llm-on-ray/pull/109) merged about add better error monitor case
  + Delete [bigdl dir&file,](https://github.com/intel/llm-on-ray/pull/198) Enable start\_serve in a common place for all test case,[Enable all openai chat apis](https://github.com/intel/llm-on-ray/pull/167)
  + Change docker2sh structure, can put most items to bash, but map in bash can't be used directly
  + [TODO]Finish docker2sh PR, List openai full api support

Aurora:

* + Pending access Aurora cluster.

WW16

=======

Ray Summit 2024: Drafted two proposals, one for Gaudi on Ray, one for LLM-on-Ray

Open:

- Enable LLama3 in llm-on-ray (CPU, Gaudi) - Lin Zhi

Status Update:

==============

LLM on Ray Serving:

* + Update to upstream vLLM: [Update vllm to use upstream v0.4.0.post1](https://github.com/intel/llm-on-ray/pull/179)
    1. PR to fix: <https://github.com/vllm-project/vllm/pull/3993>. Issues found and reported to Yuan: 1) can't run in offline mode if only model id provided. 2). Cancel exception raised after output generated
  + RayLLM upstream:
    1. Enabling CPU inference: <https://github.com/ray-project/ray-llm/compare/master...xwu99:ray-llm:support-vllm-cpu?expand=1>. Also pending on above vLLM upstream fix
  + Performance Benchmark (Keping)
    1. Align llm-on-ray with latest performance (100% CPU, reduce dashboard overhead etc.)
    2. Dynamic Batching test result (cpu, Gaudi): [dynamic batching](onenote:Keping.one#dynamic%20batching&section-id={4C5A26D5-01E6-44D8-B726-942506655618}&page-id={15251928-29FA-4B99-8A96-87A967885D02}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML)
       - [Issue about using ipex on cpu](https://github.com/intel/llm-on-ray/issues/197)
  + Benchmark visualization script: [PR#195](https://github.com/intel/llm-on-ray/pull/195)
  + Finished chat template code in llm-on-ray inference and rebase main branch and more testing.
  + Refactor
    1. Plan to be discussed: [LLM-on-Ray Improvement Plan](onenote:#LLM-on-Ray%20Improvement%20Plan&section-id=%7B08438F69-9374-47E5-83E4-DE309A4A75B0%7D&page-id=%7BE134E626-3509-4499-BFAC-1CDB1E8DB2DC%7D&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML/Plan.Status.Update.one)  ([Web view](https://intel.sharepoint.com/sites/SparkA21/_layouts/OneNote.aspx?id=%2Fsites%2FSparkA21%2FSiteAssets%2FSparkA21%20Notebook&wd=target%28Ray.ML%2FPlan.Status.Update.one%7C08438F69-9374-47E5-83E4-DE309A4A75B0%2FLLM-on-Ray%20Improvement%20Plan%7CE134E626-3509-4499-BFAC-1CDB1E8DB2DC%2F%29))
    2. Refactoring predictors and deployment flow
  + Comprehensive OpenAI API Support
    1. Gaps?
    2. Tests?
  + Investigate Model Evaluation
    1. [EleutherAI/lm-evaluation-harness: A framework for few-shot evaluation of language models. (github.com)](https://github.com/EleutherAI/lm-evaluation-harness)
    2. [open-llm-leaderboard (Open LLM Leaderboard) (huggingface.co)](https://huggingface.co/open-llm-leaderboard)

LLM-on-Ray Quantization:

* + Researched seamless integration between vllm and neural-speed, including
    1. wrapping native data in pytorch tensor
    2. support neural-speed quantization with vllm's linearmethod, also checked ipexllm's impl
    3. Manage ns kv cache in vllm
    4. Vllm extension
  + Track ns bug fixing

Gaudi:

* + Fine-tuning:
    1. Merged QA benchmark scripts for llama, [#411151](https://gerrit.habana-labs.com/#/c/411151/)
    2. Updated implementation of fine-tuning example for upstreaming #[44667](https://github.com/ray-project/ray/pull/44667)
       - Before update (GaudiAccelerator + epoch-step-loop), all execution mode can be run
       - After update (GaudiTrainer), only lazy mode can be run, eager/compile mode fail to run, issue reported [here](https://jira.habana-labs.com/browse/SW-182652).
    3. Debugging Llama-2-70b + deepspeed problem: RuntimeError: No backend type associated with device type cpu
  + Stable Diffusion Example:
    1. Created a jupyter notebook to finetune stable diffusion, sd 1.5 passes, sd2 fails. Will try in newer docker image
    2. TODO: integrate Ray train checkpoint, add generate after finetune

Aurora:

* + Pending access Aurora cluster.
  + Continue support OneDAL team to solve issues [DALL-7083](https://jira.devtools.intel.com/browse/DAALL-7803).

Tests:

* + Update bigdl 2 ipex-llm, [PR#187](https://github.com/intel/llm-on-ray/pull/187)merged
  + Fix [PR#167](https://github.com/intel/llm-on-ray/pull/167) for OpenAI protocol and router test [issue#151](https://github.com/intel/llm-on-ray/issues/151) after review
  + PR 109 rebase main
  + [TODO] **Delete bigdl dir&file**, Enable start\_serve in a common place for all test case,Enable all openai chat apis,**List openai full api support**

CI:

* + created a system user and replace my account with it in all CI hosts. Gaudi host?
  + Added cron job to relogin hourly

WW15

=======

Intel Vision

Gaudi support in Ray: <https://www.anyscale.com/blog/accelerating-ai-harnessing-intel-gaudi-3-with-ray-2-10>

OPEA: <https://github.com/opea-project/GenAIExamples>

Status Update:

==============

LLM on Ray Serving:

* + Batching: [Refactor and Enable batching for both simple and openai by xwu99 · Pull Request #153 · intel/llm-on-ray (github.com)](https://github.com/intel/llm-on-ray/pull/153)
  + Update to upstream vLLM: [Update vllm to use upstream v0.4.0.post1](https://github.com/intel/llm-on-ray/pull/179)
    1. Wait for Li Jiang to fix AsyncEngine interface in upstream
  + Refactor
    1. [Update device name in YAML files](https://github.com/intel/llm-on-ray/pull/182)
    2. TODO: refactor predictors and deployment flow
    3. Plan: [LLM-on-Ray Improvement Plan](onenote:#LLM-on-Ray%20Improvement%20Plan&section-id=%7B08438F69-9374-47E5-83E4-DE309A4A75B0%7D&page-id=%7BE134E626-3509-4499-BFAC-1CDB1E8DB2DC%7D&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML/Plan.Status.Update.one)  ([Web view](https://intel.sharepoint.com/sites/SparkA21/_layouts/OneNote.aspx?id=%2Fsites%2FSparkA21%2FSiteAssets%2FSparkA21%20Notebook&wd=target%28Ray.ML%2FPlan.Status.Update.one%7C08438F69-9374-47E5-83E4-DE309A4A75B0%2FLLM-on-Ray%20Improvement%20Plan%7CE134E626-3509-4499-BFAC-1CDB1E8DB2DC%2F%29))
    4. Refactor data type pr [PR#166](https://github.com/intel/llm-on-ray/pull/166) : fixing ci; add logic for hpu, newly added cpu(vllm, mllm, etc.) to take care
  + Benchmark
    1. Last result (KP): Ray serve wrapper is slower than vllm fastapi. 216.73/req vs 155.40/req (256 reqs)
    2. [OpenMP OMP\_PROC\_BIND=true](onenote:Performance.one#OpenMP%20OMP_PROC_BIND=true&section-id={D09FBE29-B0DE-4E09-BBB3-8333BBC71AED}&page-id={13A0EA04-7AE0-4FCA-ADF5-4808DC4075DD}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML)
    3. [Benchmark WW14](onenote:Performance.one#Benchmark%20WW14&section-id={D09FBE29-B0DE-4E09-BBB3-8333BBC71AED}&page-id={98F2A01D-CB4E-41CF-AFDE-7284FA0FCD36}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML)
  + Verify dynamic batching on Guadi
    1. todo: generate throughput metric result based on benchmark script.
  + Change ignore\_eos to a parameter in benchmark script: [PR#184](https://github.com/intel/llm-on-ray/pull/184)
  + Integrating chat template in llm-on-ray inference.

LLM-on-Ray Quantization:

* + Completed the integration of vllm and neural-speed (replaced neural-speed's simple cont. batching scheduling with vllm's scheduler), including

sync and async vllm engines.

* + Will consolidate the integration and plug it into llm-on-ray

LLM on Ray Finetuning:

* + Submit and fixed comments PR [#178](https://github.com/intel/llm-on-ray/pull/178) and merged PR [#181](https://github.com/intel/llm-on-ray/pull/181).

Tests:

* + Create [PR#167](https://github.com/intel/llm-on-ray/pull/167) for OpenAI protocol and router test [issue#151](https://github.com/intel/llm-on-ray/issues/151)
  + Merged PR 156 for Fix test error [issue#152](https://github.com/intel/llm-on-ray/issues/152),PR162 for docker build error
  + Update bigdl-llm to ipex-llm [PR#187](https://github.com/intel/llm-on-ray/pull/187)
  + PR 109 fix review

Gaudi:

* + Fine-tuning:
    1. Upstream PR: #[44667](https://github.com/ray-project/ray/pull/44667), pending on review, support 3 execution mode.
    2. QA benchmark script, developping
  + Resnet & bert benchmark script submitted, no response yet
  + Stable Diffusion & ViT:
    1. Stable Diffusion: found example in optimum habana, seems easy to integrate
    2. ViT example: use ray data to batch inference, may need to upstream changes to ray data. Habana dataloaders to explore
  + Run TGI-Guadi: [TGI Guadi Test](onenote:Keping.one#TGI%20Guadi%20Test&section-id={4C5A26D5-01E6-44D8-B726-942506655618}&page-id={8B2D2A92-8643-4FC0-9385-359523DE3588}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML)
  + Run vllm on Guadi: in progress. vllm can be successfully installed using the docker environment provided by llm-on-ray,

but an error([vllm Guadi Test](onenote:Keping.one#vllm%20Guadi%20Test&section-id={4C5A26D5-01E6-44D8-B726-942506655618}&page-id={22F5CFD5-D12F-4890-850A-608CC3B96FA0}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML)) will be reported during execution.

UI:

* + Fix webui bugs and adapt to ray2.10: [PR#186](https://github.com/intel/llm-on-ray/pull/186)

Aurora:

* + Pending access Aurora cluster.
  + Support OneDAL team to solve issues [DALL-7083](https://jira.devtools.intel.com/browse/DAALL-7803).

CI:

* + Restored self-hosted runners
  + Worked with yizhong to setup self-hosted runner in gaudi2 host

Yizhong:

* + Added an exclusion file to license check so that we can add specific files or regular expressions for files that don't need to be checked. ([PR](https://github.com/intel/llm-on-ray/pull/154) merged)
  + This PR is merged and I also fixed some CI errors and lints through this [PR](https://github.com/intel/llm-on-ray/pull/154).
  + I fixed all errors caused by removing configs from serve.py according to the discussed list of [Issue#146](https://github.com/intel/llm-on-ray/issues/146).
  + I added configs to gpt2-ci.yaml for better test coverage after some configs are removed from serve.py. ([PR](https://github.com/intel/llm-on-ray/pull/165) pending on review)
  + Now I am working on rebuilding the environment of gaudi node and will continue resolving the issues like missing llmray-build files, CI workflow canceled on Gaudi label etc. ([PR](https://github.com/intel/llm-on-ray/pull/185) pending on update)

WW13

=======

IDC Gaudi Server - Access & Demo

ssh to [jumpid@10.239.44.190](mailto:jumpid@10.239.44.190) first, then connect to the Gaudi Server .

Note:

* + Don’t use user sdp directly, create your own account
  + Let's use the same directory to store models.

|  |  |  |
| --- | --- | --- |
| **Instance Name** | **IP Address** |  |
| aise-cluster-00-0 | ssh -i ~/.ssh/id\_ed25519\_idc -J [guest@146.152.224.71](mailto:guest@146.152.224.71) [sdp@100.83.111.228](mailto:sdp@100.83.111.228) | Mt.sill (Chendi) |
| aise-cluster-00-1 | ssh -i ~/.ssh/id\_ed25519\_idc -J [guest@146.152.224.71](mailto:guest@146.152.224.71) [sdp@100.83.111.248](mailto:sdp@100.83.111.248) | Dev (Gangsheng, Zhi ) |
| aise-cluster-00-2 | ssh -i ~/.ssh/id\_ed25519\_idc -J [guest@146.152.224.71](mailto:guest@146.152.224.71) [sdp@100.83.111.243](mailto:sdp@100.83.111.243) | CI, dev (Yizhong; ) |
| aise-cluster-00-3 | ssh -i ~/.ssh/id\_ed25519\_idc -J [guest@146.152.224.71](mailto:guest@146.152.224.71) [sdp@100.83.111.236](mailto:sdp@100.83.111.236) | Perf Test (Keping, ) |

Status Update:

==============

Gaudi:

* + Finetuning
    1. Merged [#155](https://github.com/intel/llm-on-ray/pull/155), [#160](https://github.com/intel/llm-on-ray/pull/160) , supported fine-tuning on Gaudi2 by DDP and DeepSpeed.
    2. TODO: Upstream llama2 finetuning example, Multi-node support, verify 70b with Deepspeed.
  + Serving
    1. Measuring QPS and Latency of inference by using Locust.
  + Enable CI (Yizhong)
  + Test Performance of TGI-Gaudi, vLLM-Gaudi, LLM-on-Ray (Keping)

<https://github.com/huggingface/tgi-gaudi>

<https://github.com/HabanaAI/vllm-fork/tree/habana_main>

[https://github.com/intel/GenAIExamples](https://github.com/intel/GenAIExamples/blob/main/ChatQnA/README.md#environment-setup)

* + Continuous batching

**Gaudi Enabling:**

* + Serve
    1. QA follow-ups, almost done
    2. Ray train benchmark script, testing resnet, bert not yet
    3. Deepspeed inference example submitted to upstream, following-up

LLM on Ray Serving:

* + [LLM-on-Ray Improvement Plan](onenote:#LLM-on-Ray%20Improvement%20Plan&section-id=%7B08438F69-9374-47E5-83E4-DE309A4A75B0%7D&page-id=%7BE134E626-3509-4499-BFAC-1CDB1E8DB2DC%7D&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML/Plan.Status.Update.one)  ([Web view](https://intel.sharepoint.com/sites/SparkA21/_layouts/OneNote.aspx?id=%2Fsites%2FSparkA21%2FSiteAssets%2FSparkA21%20Notebook&wd=target%28Ray.ML%2FPlan.Status.Update.one%7C08438F69-9374-47E5-83E4-DE309A4A75B0%2FLLM-on-Ray%20Improvement%20Plan%7CE134E626-3509-4499-BFAC-1CDB1E8DB2DC%2F%29))
  + Batching: [Refactor and Enable batching for both simple and openai by xwu99 · Pull Request #153 · intel/llm-on-ray (github.com)](https://github.com/intel/llm-on-ray/pull/153)
  + Benchmark: debug code, Ray serve wrapper is slower than vllm fastapi. 216.73/req vs 155.40/req (256 reqs)
  + Merged [#141](https://github.com/intel/llm-on-ray/pull/141), supported google/gemma-2b inference and fine-tuning for CPU.
  + Research and investigate [transformer chat template](onenote:Minming.one#Chat%20Template&section-id={5785A99B-C365-46B8-91DD-C034A272F7BB}&page-id={E846E2A5-AB0A-4BFD-92BE-7F5A04893042}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML).

LLM-on-Ray Quantization:

* + Neural-speed model can be quantized, loaded and initialized from vllm with CPU only.
  + Dig into neural-speed's beam-search impl which is deeply coupled with its own cont. batching scheduler. WIP to decouple them

and adapt to vllm cont. batching.

Tests:

* + Tests for OpenAI protocol [issue#151](https://github.com/intel/llm-on-ray/issues/151) copy from ray-llm ,local success ,[TODO]ready to create a new pr.
  + Fix test error [issue#152](https://github.com/intel/llm-on-ray/issues/152) about single query test
  + Delete usuless"yes" on workflow yaml fix "docker container prune" error
  + [TODO] Add docker login check [self-hosted add cron]

Aurora:

* + Pending access Aurora cluster.
  + Revert oap-mllib [#376](https://github.com/oap-project/oap-mllib/pull/376) and merged [#375](https://github.com/oap-project/oap-mllib/pull/375)

WW12

=======

Mar MSR

Insights/OKR update

Github copilot

Status Update:

**Gaudi Enabling:**

* + Fine-tuning:
    1. Performance fixed, by updated data process strategy (grouping), llm-on-ray can fine-tuned more efficient than official example.
    2. Trying fine-tuning with Llama2-70b with deepspeed, got environment problems: "RuntimeError: synStatus=8 [Device not found] Device acquire failed"
       - Success running official deepspeed example, lora + llama2-70b + 8hpu + alpaca need 1hour.
       - Got some errors when integrated hpu + deepspeed to llm-on-ray.

LLM on Ray Serving:

* + Refactor and enable static, dynamic and continuous batching: [Refactor and Enable batching for both simple and openai #153](https://github.com/intel/llm-on-ray/pull/153)
  + Add openai support in benchmark script.
  + Debug benchmark performance: output length, data sampling method, the difference between vllm parameter.
  + Completed integrate google/gemma-2b inference for CPU. Continue to integrate mistralai/Mixtral-8x7B-Instruct-v0.1.
  + Research and investigate [transformer chat template](https://huggingface.co/docs/transformers/main/chat_templating).
  + Tool/function support, ready to merge.
  + TODO:
    1. Gaudi benchmarks
    2. Full support for OpenAI http protocols and validation (Q2 OKR),
       - support single request multiple prompts for OpenAI
    3. Improve benchmark script
       - separate latency & throughput benchmark
       - support static batching
    4. Docs
       - Tutorial: Llm-on-ray running on AWS from [llm-projects/ray-cluster-aws/cluster-aws.yaml at main · xwu99/llm-projects (github.com)](https://github.com/xwu99/llm-projects/blob/main/ray-cluster-aws/cluster-aws.yaml)
       - Blog on llm-on-ray benchmarks

LLM on Ray Quantization:

* + Continue studying on various continuous batching impls in vallina vllm and its variant, like bigdl vllm, as well as neural-speed's cpp impl.
  + Evaluated several ways to integrate neural-speed's quant to llm-on-ray. We decided to use vllm's cont batching variant and customized

neural-speed, to expose cpp interface as python API, as our quant solution in llm-on-ray. Moving forward, we'll make neural-speed and vllm

integration more seamlessly.

CI & Tests:

* + Merged [PR#147](https://github.com/intel/llm-on-ray/pull/147) for docker error show passed
  + Fix [PR#123](https://github.com/intel/llm-on-ray/pull/123) Move Docker Instruction to Bash in workflow\_tests, inference and finetune start wait too long in self-host ci.
    1. Gitub ubuntu ci can't serve big model and inference\finetune
  + Fix print too much output ,find key error flag in [PR#109](https://github.com/intel/llm-on-ray/pull/109) add other little models
    1. List error case
  + [TODO] Tests for OpenAI protocol [issue#151](https://github.com/intel/llm-on-ray/issues/151) , single test error fix [issue#152](https://github.com/intel/llm-on-ray/issues/152)
    1. Add Ray-llm openai test case

Aurora:

* + Pending access Aurora cluster.

WW11

=======

* + DAP passdown
    1. Passdown
       - Interim leader & Gaudi - volume customers
    2. Demos being planned
       - Manoj -> disease prediction and visual quality
       - Monica's demo -> easydata + LLM
       - Andres demo -> ray
       - Justin demo on intel vision, andres helping -> non->intel repos
         * Open source
         * Not really need to show the fine-tune data
    3. Gaudi2 HW resources
       - Manoj's team got access (for demo)
       - Tyler got two for Mt.Sill
       - Ray already have for dev. Purpose
       - 14-nodes to be ready in 3/25 (expected)
  + Open:
    1. Ref app:

Haihao team: <https://github.com/intel-sandbox/generativeaiexamples>

TCS out sourcing is also working on it…

Status Update

=============

**Gaudi Enabling:**

* + Habana Collaboration & Ray Upstream
    1. Ray Train PR merged: <https://github.com/ray-project/ray/pull/43343>
    2. Serving perf benchmark: optimum-habana benchmark scripts adapted ray. w/o ray performance matched

* + Llm-on-ray with Gaudi Demo Video: [llm-on-ray\_gaudi\_demo\_v3.mp4](https://intel.sharepoint.com/:v:/s/SparkA21/ERelJ6pJPd9ImXTjk0fiEi8BbOOPUdRPc4vC2Y2QI-_IwA?e=HkaYE3).

"Lots of really positive feedback, Sid showed the demo to Wei and multiple external/internal stakeholders and received very good feedback. SMG folks is “chasing” for the demo for externally promotion for Gaudi for now. "

Remaining issues/ARs

* + Finetuning: Single card perf:
  + Finetuning: Multi card functionality:
  + Serving: multi session doesn't work:
  + Serving: incorrect output issue
  + Review & merge PRs/Docs
  + Record a video for all Intel platforms, instead of just for Gaudi. (Carson)
  + Fine-tuning:
    1. Running benchmark of llm-on-ray vs optimum examples, llm-on-ray very slow on some steps, maybe not calling *mark\_step* [[1]](https://docs.habana.ai/en/latest/PyTorch/PyTorch_Model_Porting/Porting_PyTorch_Models_to_Gaudi.html?highlight=mark_step#importing-habana-frameworks-torch-core) [[2]](https://forum.habana.ai/t/when-to-use-htcode-mark-step/446)
    2. Fail to run fine-tuning workflow on 2 node ray cluster, the [problem](https://forum.habana.ai/t/model-to-device-faile-runtimeerror-synstatus-8-device-not-found-device-acquire-failed/1105/1) report to community, still need to debug.

Customer Engagement:

* + Longshot.AI have very good feedback on llm-on-ray: “The Inference for same model and same hardware has brought in more than 5x improvements compared to using other approaches. The amount of time spent exploring, learning and debugging while finetuning and inference has drastically reduced upto 8x.” From their roadmap, they plan to migrate to fully to Intel. <https://github.com/thequantumquirk/marp-shot>
  + CDW: Sid showed the llm-on-ray demo, Chendi also gave a live demo on Gaudi. Very good feedback.
  + Tencent: Plan to introduce LLM-on-Ray (For Xeon) and vLLM.
  + LTIMindTree: MLIT-952 Starcoder2 support: Keping verified latest Transformer source code worked.
  + LTIMindTree: MLIT-944 UI Crash: Keping is working on new UI with easy data team.
    1. Update UI, use ray submit job api to submit finetuning task and get the status of job. [(update branch)](https://github.com/KepingYan/llm-on-ray/commit/276f566c4f5e27eb86587cfb7e83fa337f226c68)
  + IDC Team: Benchmarking TGI, but wanted to move to Ray. They will test llm-on-ray on SPR first. We will merge Gaudi PRs and provide them next week for testing.

We will share our benchmark result on SPR. They will share their result with TGI.

They also mentions interests in: <https://github.com/neuralmagic/deepsparse> <https://github.com/intel/neural-speed>

LLM on Ray Quantization:

* + Built and studied vllm-cpu branch
  + Compared vanilla vllm, vllm-cpu, vllm with bigdl-llm and potential vllm with ITREX. Summarized in [onenote](onenote:Jiafu.one#Continuous%20Batching%20Impls.&section-id={4D055AC1-38BB-4EC5-9104-37896B146682}&page-id={59FFA48F-B71E-4D29-8AB6-76AAF892D960}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML).

LLM on Ray Serving:

* + Benchmark: [Results](https://intel.sharepoint.com/:x:/r/sites/SparkA21/_layouts/15/Doc.aspx?sourcedoc=%7BB3960FE7-8881-4BBB-90CE-F7ED7547CFBE%7D&file=LLM.on.Ray.Perf_Throughput.xlsx&action=default&mobileredirect=true), [Slides](https://intel.sharepoint.com/:p:/r/sites/SparkA21/_layouts/15/Doc.aspx?sourcedoc=%7B25CDA347-6640-457B-A97F-7C0D51C08CA3%7D&file=LLM.on.Ray.Perf.pptx&action=edit&mobileredirect=true).
  + Llm-on-ray running on AWS setup config: [llm-projects/ray-cluster-aws/cluster-aws.yaml at main · xwu99/llm-projects (github.com)](https://github.com/xwu99/llm-projects/blob/main/ray-cluster-aws/cluster-aws.yaml)
  + TODO: Checking batching on Ray Serve, improve benchmark script (seperate latency & throughtput benchmark)

LLM on Ray Fine-tuning:

* + Submit two PRs for the projects 'google/gemma-2b' and 'mistralai/Mixtral-8x7B-Instruct-v0.1', which are currently pending review.

CI & Tests:

* + Merged [PR#83](https://github.com/intel/llm-on-ray/pull/83)
  + Fix workflow bash set -e pipefail error in [PR#147](https://github.com/intel/llm-on-ray/pull/147) (Docker build error but pass ,and show error in start docker?)
  + Sync [PR#123](https://github.com/intel/llm-on-ray/pull/123) basic functions Move Docker Instruction to Bash in workflow\_tests, Inference and finetune has some error need self-hosted
    1. Try if github ubuntu can inference or finetune big model
  + **[IM]**Tests details case martix about error case on [PR#109](https://github.com/intel/llm-on-ray/pull/109) and openai agreement test
    1. [TODO]fix print too much output ，find key error flag

Aurora:

* + Pending access Aurora cluster.
  + Completed oap-mllib OpenSSF issues and awaiting the results.
  + Meeting with mohamod. He said all aurora compute nodes are taken for IO 500 till end of May.

WW10

Passdown:

China LLM app: llm-on-ray, vllm, mt.sill

LLM-on-Ray with Gaudi demo recording

Status Update

=============

Habana + Ray project:

* + Fine-tuning:
    1. Updated [PR#131](https://github.com/intel/llm-on-ray/pull/131/files), integrated fine-tuning on Gaudi2 with reuse current codes.
    2. Support UI fine-tuning demo on Gaudi2
    3. Debugging low training speed on DDP mode with multi HPUs.

Inference/Serving

* + Merged [PR#106](https://github.com/intel/llm-on-ray/pull/106), streamline directory structure (moved all code to llm\_on\_ray), new way to serve and finetune:
    1. llm\_on\_ray-serve --config\_file llm\_on\_ray/inference/models/gpt2.yaml
    2. llm\_on\_ray-finetune --config\_file llm\_on\_ray/finetune/finetune.yaml
  + benchmark: [Experiments](onenote:Performance.one#Experiments&section-id={D09FBE29-B0DE-4E09-BBB3-8333BBC71AED}&page-id={3F3D7F15-B4CB-4538-83C5-65F8240435B8}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML). in progress.

LLM on Ray Quantization:

* + Reproduced perf data of llama2-7b in neural-speed's [blog](https://medium.com/@NeuralCompressor/llm-performance-of-intel-extension-for-transformers-f7d061556176) with both their perf script and ITREX API
  + Further study on neural-speed's design and impl. Their perf gain attributes to AMX assembly code, MHA and FFW ops fusion.

CI & Tests:

* + Fixing [PR#83](https://github.com/intel/llm-on-ray/pull/83) after PR106 merged
  + Finish [PR#123](https://github.com/intel/llm-on-ray/pull/123) Complete the basic functions Move Docker Instruction to Bash in my own small project and sync to [PR#123](https://github.com/intel/llm-on-ray/pull/123) today
  + [IM]Tests details case martix about error case on [PR#109](https://github.com/intel/llm-on-ray/pull/109) and openai agreement test
  + [TODO]Docker build error but pass ,and show error in start docker?

Aurora:

* + Completed benchmark OneDAL PCA/Cor/Sum. Both K-Means and PCA exhibit significant performance degradation.

图表, 折线图

描述已自动生成

* + When executing OAP PCA/Cor/Sum with multiple nodes, oneCCL will error.
  + When executing oneDAL PCA with multiple nodes, it frequently tends to crash.
  + Continue fix OpenSSF issues.
  + Setup DAOS servers in the 10 nodes with stable DAOS version. Then verified DAOS worked with both hadoop FS and remote shuffle plugin

In HiBench repartition workload.

WW09

DCAI Org Change

Efficiency Improvement: iGPT, Copilot

Status Update

=============

Habana + Ray project:

* + Fine-tuning:
    1. validated the examples of fine-tuning Llama2 on HPU, with single HPU and multi HPUs
    2. validated the fine-tuning Llama2 on HPU by our llm-on-ray workflow, with single HPU and without ray
    3. integrating llm-on-ray with multi HPUs and ray, got failure message "Habana device not initialized."
    4. validated the resnet example by linzhi
  + Ray Train:
    1. PR comments addressed
  + UT PR:
    1. Pending on more review

Inference/Serving:

* + Benchmark script completed
    1. Added ipex prompt dataset and synthetic prompts
    2. Save performance results and individual request/response to files
    3. TODO: improve error handling and supported parameters
  + vLLM CPU update: Yuan's team is working on upstreaming code to vLLM, they also engage with IPEX team
  + Reproduce previous performance and run benchmark script: [run QPS](onenote:Performance.one#run%20QPS&section-id={D09FBE29-B0DE-4E09-BBB3-8333BBC71AED}&page-id={866AD261-8225-4EBF-90CD-F628AD2BC8BC}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML).
  + Fix package path: [PR#106,](https://github.com/intel/llm-on-ray/pull/106) review in progress.

CI & Tests(Yutianchen):

* + Fixing [PR#83](https://github.com/intel/llm-on-ray/pull/83) after 0229review
  + Start a new [PR#123](https://github.com/intel/llm-on-ray/pull/123) for move all docker related blocks for all workflow actions into functions in some bash script
  + Tests details case martix about error case on [PR#109](https://github.com/intel/llm-on-ray/pull/109) and openai agreement test
  + Copilot has pay error due to government restrictions. Use codegeex replace
  + [Test Framework] Add test scripts for query\_single.py [PR#93](https://github.com/intel/llm-on-ray/pull/93)
  + [Test Framework] Add test scripts for serve.py [PR#92](https://github.com/intel/llm-on-ray/pull/92)
  + [Test Framework] Add test scripts for getting started and setup [PR#84](https://github.com/intel/llm-on-ray/pull/84)

LLM on Ray UI:

* + Sync with easydata team. Will integrate deploy and inference modules into their web ui.
  + Add an option to disable SSH. Llama2 7b working on UI now.
    1. CPU metrics not working now
    2. 70b not tested yet. 144 not enough devices

LLM on Ray Quantization:

* + Studied ITREX and neural-speed in terms of quantization. It turned out they use NE format instead of GGUF format to get huge perf gain over llama.cpp.
  + Reproduced their perf data with a different benchmark script than run\_tp.sh. Will discuss with weixin this afternoon.
  + Get better understanding of llama.cpp 's compute graph and their parallel computation. It'll help my neural-speed study.

Aurora:

* + Run oap kmeans on 64 nodes with new generate dataset.

图表, 折线图

描述已自动生成

* + Fixed oap-mllib, raydb and romte-shuffle openSSF and CVE issues

<https://github.com/oap-project/remote-shuffle/pull/76>

<https://github.com/oap-project/oap-mllib/pull/371>

<https://github.com/oap-project/oap-mllib/pull/370>

<https://github.com/oap-project/raydp/pull/401>

<https://github.com/oap-project/raydp/pull/400>

* + Prepared PCA, Cor and Sum data with generate data script on Aurora.
  + Investigated OpenSSF (formerly CII) Best Practices Badge.

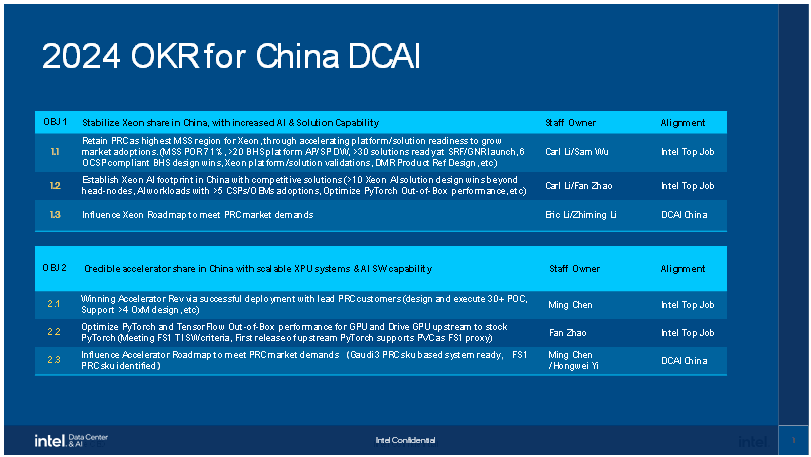
WW08

* + Refined OKR for DAP

AISE KR3.3.4 
AISE KR3.3.5 
Gaudi2/PVC enabling in Ray and top-of-the stack reference- 
apps/workflows 
(Sid) 
Provide extensible, ready-built reference APPs with E2E 
workflows on PVC/Gaudi2 (aka. Mt.Sill) 
(Sid/Manoj/Huma/Ke) 
AMB: 
1) Upstreaming Gaudi2/PVC accelerator support to Ray community 
2) Ray based examples demonstrating LLM fine-tuning & inference workflows and 
reference APPs for PVC/Gaudi2 
3) Deliver comparable performance 
AMB 
- 4 PVC/Gaudi2 ready ref-APPs and 10 refkits delivered 
- 3 customer POCs/Design wins 
DCAI 
SMG 

* + Ray roadmap was included in the references\_kit

REFERENCES 2024 ROADMAP (DRAFT) 
QI'24 
Mt.Sill and Ray (4+1) 
Ray-based LLM workflows on 
Gaudi2 
Upstream Gaudi2 support for Ray 
Serve 
Ref appl on Xeon and Gaudi2 
Ref app2 on Xeon and Gaudi2 
Mt.Whitney (5 references) 
No-change 
Apollo toolkits (10 refkits) 
Port 2 to GPU and validate it 
Q2'24 
Mt.Sill and Ray 
Ray-based LLM workflows on PVC 
Upstream Guaid2 support in Ray 
Train 
• Ref appl on PVC 
• Ref app2 on PVC 
• Ref app3 on Xeon and Gaudi2 
• Ref app4 on Xeon and Gaudi2 
Mt.Whitney (5 references) 
No-change 
Apollo toolkits (10 refkits) 
• Port 3 to GPO and validate it 
Q3'24 
Mt.Sill and Ray 
Ray-based LLM workflows on 
Gaudi2 
Upstream PVC support in Ray Serve 
• Ref app4 on PVC 
Ref apps on Xeon and Gaudi2 
Mt.Whitney (5 references) 
No-change 
Apollo toolkits (10 refkits) 
• Port 2 to GPLJ and validate it 
Q4'24 
Mt.Sill and Ray 
Ray-based LLM workflows on 
Gaudi2 
Upstream PVC support in Ray Train 
Ref app5 on PVC 
Mt.Whitney (5 references) 
Discontinue Mt.Whitney 
Apollo toolkits (10 refkits) 
• Port 3 to GPO and validate it 
•Q2-Q4 focus will be updated according to customer request 

* + DCAI China OKR
  + 

* + 2024 OKR for China DCAI 
    mc osv'EV 
    Les n 
  + AISE F2F passdown from Fan & Sid
    1. Fan - More Top-of-Stack work - but how to align with PRC customers
    2. Sid - Repurpose of ITREX/Neuralchat/LLM-Ray/BigDL-LLM (still ongoing)
  + BDF internal discussion - Innovation
    1. Innovation & path-finding new AI opportunities
  + Proposal on how to improve efficiency
    1. iGPT
    2. Copilot
    3. Github code review bot

Status Update:

LLM on Ray Serving:

* + Studied ipex quantization and neural-speed optimized llamacpp
  + Fix package path: [PR#106,](https://github.com/intel/llm-on-ray/pull/106) need to simplify execution commands, python -m llm\_on\_ray.finetune.finetune -> python -m llm\_on\_ray-fineutne
  + Fix openai sdk response: [PR#117](https://github.com/intel/llm-on-ray/pull/117).
  + Fix gpt-j-6b serving issue: [PR#115](https://github.com/intel/llm-on-ray/pull/115).
  + Add replica number parameter: [PR#116](https://github.com/intel/llm-on-ray/pull/116).
  + Troubleshoot mpt-7b-bigdl serving issue
  + Update install / doc for installing ipex & torch-ccl using --extra-index
  + Performance: polish benchmark script, added ipex dataset for fixed input\_tokens, TODO: add first token/next token latency
  + Studied AWS EC2: m7i instances for SPR (192 vCore max). TODO: how to create Ray cluster and auto scale instances with Ray in AWS

New model support: Mistral 7\*8b, google gemma

LLM on Ray UI:

* + Review on POC functions with easydata team.
  + Review MLLM [PR#107](https://github.com/intel/llm-on-ray/pull/107) and make some changes on UI.

CI & Tests(Yutianchen):

* + Fixing [PR#83](https://github.com/intel/llm-on-ray/pull/83) after review
  + Tests details case martix about error case on [PR#109](https://github.com/intel/llm-on-ray/pull/109)
  + More utils & predictor tests, processing on [PR#76](https://github.com/intel/llm-on-ray/pull/76); detail test plans on wiki [Test Framework Design](https://wiki.ith.intel.com/display/HPDA/Test+Framework+Design)
  + [TODO] openai agreement test, use Copilot set up ut tests

Habana + Ray project:

* + Resolved the hanging issue. Currently facing a Rank Error during the allreduce process in the data parallel group. Reported to Wuyuan.
  + Finished the [Habana Cluster Usage Guide](https://wiki.ith.intel.com/display/HPDA/Habana+Cluster+Usage+Guide), the build wheel pipeline, and the QA documentation for ResNet/BERT training.
  + Ray Serve Example PR #42820 and its follow-up #43304 has been merged. Deepspeed inference PR to be submitted.: only support single node; multi-node not tested
  + Ray train & UT PR is submitted.
  + HPU DeepSpeed predictor in llm-on-ray is ready to review. Added torch\_dist.
  + Setup environment and familiar with usage of Habana vm machine.
  + Benchmarking inference performance, using config of optimum-habana.

Aurora:

* + Completed oap and oneDAL K-Means benchmark from 1 to 64 nodes with new release oneapi 2023.12.15.001 on Aurora. When running on 32 nodes, OAP has a little gap with oneAPI. And when running on 64 nodes, oap has better than oneDAL. Next, we will research and investigate it.

图表, 折线图

描述已自动生成

* + Prepared PPT and reported to Taylor.

WW05

|  |  |  |
| --- | --- | --- |
|  | End of Year vacation Plan | Comments |
| Carson | 2/9 |  |
| Lin Zhi | 2/9 |  |
| Gangsheng | 2/7, 2/8, 2/9 |  |
| Xiaochang | 2/6-2/9 |  |
| Keping | 2/9, 2/18 | wfh from 2.4 |
| Tianchen | From 2/4 online |  |
| Jiafu | 2/6, 2/7, 2/8, 2/9 |  |
| Jian2 | 2/6, 2/7, 2/8, 2/9 |  |
| Yizhong | 2/18, 2/21, 2/22, 2/23 |  |
| Minming | 2/7, 2/8, 2/9, 2/18, 2/19, 2/20 |  |

Customer Update:

* + Quickheal: codellama13b + mistral-7b: 53 ms (user's ITREX LLM runtime) vs. 58 ms (llm-on-ray with bigdl-llm).
    1. End to end 50s is slower than their 30s.
    2. Need data to answer how many AWS nodes are required to support for example 100 or 200 concurrent users.
  + India AI: Sqlcoder model, e2e 28s (with bigdl).
    1. CPU utilization is lower, about 70%.
    2. Concurrent.
  + IDC team: Falcon, MPT doesn't work with Gaudi using openAI API.

* + LLM on Ray Serving:
    1. Benchmark: reproducing using vllm's api server
    2. PR review: Need to address comments.
    3. Testing: Adding setup/getting started testing to CI
    4. Discussion: new package structure. [Fix pip install path issue for certain packages by xuechendi · Pull Request #90 · intel/llm-on-ray (github.com)](https://github.com/intel/llm-on-ray/pull/90)
    5. Verified and opened two issues ([#10053](https://github.com/intel-analytics/BigDL/issues/10053), [#10054](https://github.com/intel-analytics/BigDL/issues/10054)), in BigDL

* + LLM on Ray Finetuning:
    1. Update changes to support fine-tuning on Intel GPU, [PR](https://github.com/intel/llm-on-ray/pull/88) merged
       - Support gradient-checkpointing
       - Support tensorboard tracking
    2. Run fine-tuning Llama2-7b with dataset viggo
       - Preprocessed the dataset's format to fit our workflow.
       - Checking the loss curve with Xinyao
    3. Re-enabling the GPU workflow in CICD.

* + CI & Tests(Yutianchen):
    1. Finish gpt2's http\_requests tests & openai\_sdk tests on [PR#83](https://github.com/intel/llm-on-ray/pull/83)
    2. Change tests framework from self-hosted to github ci; Use two methods: bare metal and Docker[PR#83](https://github.com/intel/llm-on-ray/pull/83)
    3. More utils & predictor tests, processing on [PR#76](https://github.com/intel/llm-on-ray/pull/76); detail test plans on wiki [Test Framework Design](https://wiki.ith.intel.com/display/HPDA/Test+Framework+Design)
    4. Tests details case martix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| parms | api\_base | model\_name | streaming\_response | max\_new\_tokens | temperature | top\_p\_values |
| default | "<http://localhost:8000/v1>" | gpt2 | False | None | None | None |
| others |  | llama-2-7b-chat-hf | True | 128 | 0.8 | 0.7 |
|  | … | … |  | … | … | …. |
| **Wrong input assert** | **Undeployed API** | **Error/none exist model name** | **Not bool variable** | **>model support or others** | **>1 or <0 or others** | **>1 or <0 or others** |

* + Finish basic parms tests;
  + Accurate assert for wrong output done on sr239 ,Prepare to submit a new PR;
  + How to test other big model on github ci **[TODO]**

* + Pretrain : to test multi-nodes.

* + Habana + Ray project:
    1. implemented a ut for Ray task and updated BERT/ResNet training scripts to Jupyter notebook format
    2. submitted PRs to the Ray repo, including unit tests and Jupyter notebooks for training BERT/ResNet models on HPU
    3. Refactor the scripts to address comments from anyscale, update doc. Synced to internal repo

* + Aurora:
    1. Hang issue:

[[TCMLIB-182] System.load TBB hang for oneAPI 2024. related to TCM - IT JIRA (Supports IC/ITS Data) (intel.com)](https://jira.devtools.intel.com/browse/TCMLIB-182)

* + Export FI\_UNIVERSE\_SIZE=1024 and FI\_CXI\_DEFAULT\_CQ\_SIZE=131072 to fix the issues that when running with 32 nodes will be crashed.([Intel® MPI Library Release Notes for Linux\* OS](https://www.intel.com/content/www/us/en/developer/articles/release-notes/mpi-library-release-notes-linux.html))
  + Completed K-Means benchmark from 1 to 64 nodes on Aurora.

图表, 折线图

描述已自动生成

* + Report scaling result for OneDAL and they send Kmenas scaling result.

|  |  |
| --- | --- |
| 图表, 折线图  描述已自动生成 | 图表, 折线图  描述已自动生成 |

* + Fixing oap-mllib [PR](https://github.com/oap-project/oap-mllib/pull/368) comments.

WW04

* + CI & Tests
    1. Run project deploy workflow, add scripts test\_getting\_started.sh and test\_setup.sh for test and one step deploy. ([PR](https://github.com/intel/llm-on-ray/pull/84/files) pending on update)
    2. Update wiki [Test Framework Design](https://wiki.ith.intel.com/display/HPDA/Test+Framework+Design), added more details about inference part and recently plan.
    3. Working on testing serve.py, query\_single.py, listed all 16 parameters, finding all possible values and trying different combinations of them.
    4. Continue updating PR [[Inference] Fix auth token and add models starcoder and llama2](https://github.com/intel/llm-on-ray/pull/39), blocked on CI proxy, failed to checkout repo.

(Yutianchen):

* + Update sr239 to ubuntu2204, set up environment
  + More detail test plans on wiki [Test Framework Design - High Performance Data Analytics Team - Intel Enterprise Wiki](https://wiki.ith.intel.com/display/HPDA/Test+Framework+Design)
    1. query\_http\_requests test framework done on sr239 servers, processing on [PR#83](https://github.com/intel/llm-on-ray/pull/83)
    2. query\_openai\_sdk test framework done on sr239 servers
  + Tests details case martix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| parms | api\_base | model\_name | streaming\_response | max\_new\_tokens | temperature | top\_p\_values |
| default | "<http://localhost:8000/v1>" | gpt2 | False | None | None | None |
| others |  | llama-2-7b-chat-hf | True | 128 | 0.5 | 0.7 |
|  |  | neural-chat-7b-v3-1 |  | 1024 | 0.8 | 0.9 |
|  | … | … |  | … | … | …. |
| **Wrong input assert** | **Undeployed API** | **Error/none exist model name** | **Not bool variable** | **>model support or others** | **>1 or <0 or others** | **>1 or <0 or others** |

* + Other parms tests to add
  + Accurate assert for wrong output **[TODO]**
  + Self-host ci using problem
    1. Testing the query requires using a self host machine Docker CI
    2. There seems to be some issues with the self host machine network

* + LLM on Ray Docker Installation

Updated doc ([#77](https://github.com/intel/llm-on-ray/pull/77)) and changed way of setuptools finding packages ([#81](https://github.com/intel/llm-on-ray/pull/81))

* + LLM on Ray Finetuning:
    1. Debugged the too much memory usage problem:
       - Modified the dataset preprocess strategy, this modification can slightly improve the memory usage, while not affecting the training runtime.
       - Enabled the input gradient related features of pretrained transformers model, this upgrade can significantly enhance memory utilization, while also substantially affecting the training runtime.

* + LLM on Ray Serving:
    1. Benchmark:
       - reproduce vllm performance on SPR (Throughput: 0.96 requests/s, 460.78 tokens/s), test llm on ray performance in simple mode (Throughput: 0.45 requests/s, 215.81 tokens/). I will debug the configuration parameters of ray, such as `max\_concurrent\_queries`.
    2. Add vllm for openai api support: PR submitted, continue to refactor openai completions api.
    3. Quantization: GPTQ, AWQ, GGUF/GGML support (vllm (load), bigdl-llm (convert & load))
       - <https://huggingface.co/TheBloke>

|  |  |
| --- | --- |
| GPTQ | 1231 |
| AWQ | 1014 |
| GGUF | 1067 |
| GGML | 396 |

* + Add HPU deepspeed support: Implemented according to optimum-habana, aim to reproduce performance shown there
  + Codellama-13b support in llm-on-ray[bigdl-cpu] and benchmark

* + Habana + Ray project:
    1. QA documentation for Habana has been added, which includes environment setup, execution commands, and results.
    2. Successfully tested multi-node training for ResNet and BERT, with the training logs provided in the PR.
    3. Figured out how to deploy 70b model on habana cluster, model is available under /mnt/weka/data. QA doc is uploaded and PR is ready
    4. Working on enabling hccl for ray.air.util.torch\_dist.init\_torch\_dist\_process\_group
  + Aurora:
    1. Completed oap-mllib [PR](https://github.com/oap-project/oap-mllib/pull/368) that update oneapi to 2024.0.0, next fix comments.
    2. Investigated K-Means's result. Detail info [Benchmarking result on Aurora](onenote:Minming.one#Benchmarking%20result%20on%20Aurora&section-id={5785A99B-C365-46B8-91DD-C034A272F7BB}&page-id={CD895BFC-BE8C-45D8-898B-61CD19009854}&end&base-path=https://intel.sharepoint.com/sites/SparkA21/SiteAssets/SparkA21%20Notebook/Ray.ML).
    3. OneDAL team's plan.

图形用户界面, 应用程序

中度可信度描述已自动生成

WW03

* + Potential LLM-on-Ray Users:
    1. India AI team (Anand/Ritik): Natural language to SQL use case. Running defog/sqlcoder model using llm-on-ray and compare with their TGI solution.

TGI takes around 80-90 seconds and Initially LLM-on-Ray took about 112 seconds. After applying numa binding

and our configuration tuning, latest LLM-on-Ray result is 50 seconds.

* + QuickHeal:
    1. Running Codallama-13B- instruct on M7i.12xlarge. Target 30 ms/token, 30 sec(2k token i/p). Current: 53 ms/token for smaller prompts(512 or less)
    2. Running Mistral-7b-OpenOrca on M7i.12xlarge. Target 30 sec (2695 input tokens)
    3. Future work: Help QuickHeal to run the llm-on-ray on AWS, achieve same performance 53 ms/token first, showcase throughput/scalability value from llm-on-ray

* + Topic collection for AISE F2F
    1. Generative BI/analytics – deliver a new application for BI leverage LLM <https://wiki.ith.intel.com/pages/viewpage.action?pageId=3390550360>
    2. Narrative SQL – natural language to SQL. (Extend Anand’ team’s work?)
    3. Cost-efficient serving solution with vLLM on Ray
    4. Please continue to add what ever you have in your mind..

* + LLM on Ray Finetuning:
    1. Running fine-tuning training runtime statistic on Intel GPU and A100, with different configurations. [Result](https://intel.sharepoint.com/:p:/r/sites/SparkA21/Shared%20Documents/General/LLM/LLM%20on%20Ray%20fine-tuning%20performance%20on%20PVC.pptx?d=w6f15075c5ab94e5bb390d419fb5137f5&csf=1&web=1&e=4Kv3M5).
    2. Try to explain why our workflow faster than LLaMA-Factory, and need more GPU memory:
       - Training time: our workflow faster 30%. different tokenizer strategy and 5% dataset as evaluation data.
       - GPU memory: our workflow need 3x memory. different tokenizer strategy.
    3. Run successfully Accelerate DeepSpeed finetuning with EleutherAI/pythia-2.8b on PVC.

* + LLM on Ray Serving:
    1. vLLM integration: PR merged. Adding benchmark scripts today. Start benchmark from next week.
    2. bigDL integration:

Ran some test on bigdl continuous batching. Debugged and reported an issue with llama-7b-chat-hf.

* + More model support: #39 Please remove token and merge.
  + Multiple prompts support: pr is ready for simple mode, will submit a new pr for openai mode.
  + Deepspeed optimization: CI passed, ready to review. I think we need to discuss whether we need to do this, because the latency might not be crucial; and vllm distributed may be more popular
  + UI
    1. Rag update: review in progress.
    2. fix log level debug.
    3. gradio is bumped to 4.11, but it can't run when gradio > 3.36

* + Habana + Ray project:
    1. Llama-70b inference: Last week tested; this week working on fixing 144's env; 7b model tested on habana cluster; not enough disk space for 70b model
    2. RN50, BERT training:
       - identified the cause of the local\_rank error
       - Test RN50, BERT training in multi-node: has built a Ray cluster across different nodes using containers

* + Improvements & Code refactor:
    1. [Document] Refactor and move source oneCCL to ray start section ([PR#44](https://github.com/intel/llm-on-ray/pull/44) merged)
    2. [Inference] Add models CodeLlama-7b and falcon-7b ([PR#12](https://github.com/intel/llm-on-ray/pull/12) merged)
    3. [Inference] Add model starcoder and enable llama2 ([PR#39](https://github.com/intel/llm-on-ray/pull/39) pending on review)

* + CI & Tests:
    1. Merged UT Tests skeleton code
    2. Supplementing wikis for some future tests plans [Test Framework Design - High Performance Data Analytics Team - Intel Enterprise Wiki](https://wiki.ith.intel.com/display/HPDA/Test+Framework+Design)

* + Aurora:
    1. Fixed oap-mllib loading libMLlibDAL.so will be blocked. Before loading so file to preload System.loadLibrary("onedal\_thread").
    2. Completed K-Means benchmark on Aurora. Next double check results.

图表, 折线图

描述已自动生成

* + Closed JIRA [DAALL-7535](https://jira.devtools.intel.com/browse/DAALL-7535), We double checkout it with the latest OneAPI that we found that the results were consistent without using "export ZE\_FLAT\_DEVICE\_HIERARCHY=COMPOSITE" about 31 Sec.

WW02

Gaudi Enabling in Ray: core, serving, finetuning,

* + vLLM Integration:

* + LLM on Ray Finetuning:
    1. PVC: finetuning LLama2, alpaca, 3 epochs, LoRA. (A100: ~2.4 hours, PVC: estimate ~3.4 hours): Use BF16/FP16, upgrade oneAPI.
    2. Checked the truly performance on one tile

* + LLM on ray serving:
    1. UT: Have some formatting issues after format check, not merge yet.
    2. Change MODEL\_TO\_SERVE env to command line argument to specify models.
    3. Continuous batching and tensor parallelism research in vllm, as well as other new trends in serving, like caching chat history, scaling llm serving in cloud

* + LLM on ray lint: Merged. Need to run format.sh before every commit.

* + Habana + Ray project:
    1. Based on Jerome's previous PR, support for HPU has been added to the Ray train module.
    2. Testing code for the ResNet and BERT models has been added and successfully executed.
    3. Serving: Prepared a demo of llama2-70b inference using 8 HPU with deepspeed. We need to set RAY\_EXPERIMENTAL\_NOSET\_HABANA\_VISIBLE\_MODULES to 1

* + Aurora:
    1. Communicate with OneDAL to rebuild the latest package on Aurora.
    2. Benchmarking k-menas on Aurora. When running oap-mllib kmeans, Loading so file will be blocked.
    3. Created new JIRA [DAALL-7600](https://jira.devtools.intel.com/browse/DAALL-7600) that k-menas benchmark with 32 nodes it will be crashed. Next verify with new oneDAL or intel mpi .
    4. WIP: Will talk to DAOS team to get some server nodes next week

* + Add models and refactor
    1. [Refactor] Refactor install oneapi and start ray cluster scripts ([PR#45](https://github.com/intel/llm-on-ray/pull/45) merged)
    2. [Refactor] Enable help and code refactor on several scripts ([PR#43](https://github.com/intel/llm-on-ray/pull/43) merged)
    3. [Document] Refactor and move source oneCCL to ray start section ([PR#44](https://github.com/intel/llm-on-ray/pull/44) pending on review)
    4. [Inference] Add models CodeLlama-7b and falcon-7b ([PR#12](https://github.com/intel/llm-on-ray/pull/12) pending on review)
    5. [Inference] Add model starcoder and enable llama2 ([PR#39](https://github.com/intel/llm-on-ray/pull/39) pending on update)

WW01

Q1 Direction:

LLM-on-Ray:

Serving:

- vLLM CPU Integration

- Quantization support (INT4, INT 8 support by BigDL or llama.cpp or others)

- larger model support on PVC/Gaudi2

- Performance validation and report. (align SPR, PVC, Gaudi performance)

- More model support

- More parameter support in Rest API.

Finetuning:

- Gaudi2 support

- Deepspeed support

- Performance, evaluation on PVC/Gaudi2 (LLAMA2 + Alpaca, Mistral +SlimOrca)

- Show finetuning benefit.

- Support OpenAI format?

- More model support

Gaudi Enabling in Ray: core, serving, finetuning,

* + vLLM Integration: generate works. WIP support streaming and openAI API

* + LLM on Ray CI: Enabled on llm-on-ray repo.

* + LLM on Ray Finetuning:
    1. PVC: finetuning LLama2, alpaca, 3 epochs, LoRA. (A100: ~2.4 hours, PVC: estimate ~3.4 hours): Use BF16/FP16, upgrade oneAPI.
    2. Added mixed\_precision parameter to support BF16/FP16

* + LLM on ray serving:
    1. Rest API merged.
    2. Disabled IPEX for neural chat model.
    3. UT: Added a few simple tests.
    4. Make ui demo module independent: <https://github.com/intel/llm-on-ray/pull/29>

* + Habana + Ray project:
    1. Based on the new PR, confirm the correct assignment of actors to HPUs
    2. Add hpu advance unit tests. Make sure the environment variable `HABANA\_VISIBLE\_MODULES` matches the HPU IDs and ensure that actors using HPUs within a placement group are scheduled correctly
    3. Add a readme of the tests
    4. Serving: Demonstrated a single process, single card example to habana. Look into deepspeed now

* + Aurora:
    1. OneDAL support default behavior-> tiles as devices.
    2. WIP Verify DAALL-6036 and DAALL-7264. Report DAALL-7264's new error to OneDAL. Next verify DAALL-6036 on Boris.
    3. Will talk to DAOS team to get some server nodes next week

LLM on Ray Serving:

* + benchamrk: