Proposal for Asia Super Computing (ASC) 2024

ZIQIN LI*, School of Cyber Science and Technology, Sichuan University, China
PENGCHAO WANG, School of Cyber Science and Technology, Sichuan University, China
JIACHEN WU, School of Business, Sichuan University, China
XINYU ZHANG, School of Cyber Science and Technology, Sichuan University, China
JUNYU ZHOU, School of Cyber Science and Technology, Sichuan University, China
KAI GAO, School of Cyber Science and Technology, Sichuan University, China

1 INTRODUCTION OF THE UNIVERSITY DEPARTMENT ACTIVITIES IN SUPERCOMPUTING

Supercomputing-related hardware and software platforms. Being a comprehensive university, Sichuan University (SCU) has several HPC platforms, hosted by different departments/schools and running multi-disciplinary workloads. For example, the Med-X Center for Informatics is a joint platform[1] co-managed by the West China Hospital and the School of Computer Science. The center has the largest HPC platform in SCU, with 103 nodes and more than 7, 200 CPU cores, reaching a storage of 14 PB and a peak performance of 1.17 PFLOPS. Other platforms include the Dawning HPC cluster of the Institute of Atomic and Molecular Physics, the Cloudkirin HPC cluster of the School of Chemical Engineering, the numerical simulation platforms of the School of Aeronautics and Astronautics, etc. The team members are from the School of Cyber Science and Engineering, which has a CPU-intensive cluster and an AI cluster. Unlike typical AI clusters that use Nvidia GPUs, the AI cluster of our department uses chips from Cambricon, a domestic AI chip vendor.

Supercomputing-related courses, training, and interest groups. Prof. Gang Chen from the School of Mathematics teaches an undergraduate course named "High-performance Computing". Other than that, several schools offer the "Computer Architecture and Organization" course and "Parallel Computing" course. There has not been an official interest group for high-performance computing at this point.

Supercomputing-related research and applications. There are multiple HPC research groups from the School of Computer Science and the School of Software Engineering. The university is also pushing very hard for multidisciplinary HPC research, such as bio-informatics, computational chemistry, and large-scale aerospace numerical simulations.

A brief description of the key achievements in supercomputing. Sichuan University hosted the 24th IEEE International Conference on High-performance Computing & Communications (HPCC, CCF-C) in 2022.

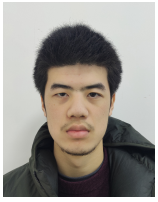
[1] http://www.wchscu.cn/resources/computer.html

2 TEAM INTRODUCTION

This is your life. Do what you love, and do it often.

Fig. 1. Team setup (Left to right: Ziqin Li, Xingyu Zhang, Pengchao Wang, Jiacheng Wu, Junyu Zhou, Kai Gao).

The team consists of 5 highly enthusiastic students who are interested in building computer systems of high performance. Ziqin, Pengchao, Xinyu and Junyu are from the School of Cyber Science and Engineering, while Jiacheng is from the School of Business. The human resources are divided as follows: Ziqin is the team leader and leads the LLM inference challenge, Junyu leads the OpenCAEPoro challenge and also manages the hardware and software infrastructure, Pengchao works on the mathematical aspects of the OpenCAEPoro challenge, Xinyu works on the LLM inference challenge, and Jiacheng works on the HPL/HPCG and helps with the OpenCAEPoro challenge.

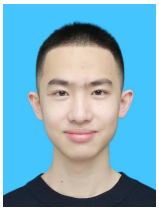


Ziqin Li is currently a sophomore at Sichuan University, majoring in cyberspace security. His research interests focus on operating systems, computer architecture, and system security. He's also in the Tencent Project Supernova.

Junyu Zhou is in his 3rd grade at the School of Cyber Science and Engineering, Sichuan University. He used to be a CTF player majoring in binary exploitation (PWN) and now expands his interests to system security and computer architecture. He is also a participant in the "one student one chip" project (https://ysyx.oscc.cc).

Pengchao Wang is now a sophomore at Sichuan University, majoring in Cybersecurity in the School of Cyber Science and Engineering. He studies with great enthusiasm and has the highest GPA constantly in his class. He has been interested in computer science and algorithms since elementary school. Pengchao was a National Olympiad in Informatics Winter Camp bronze medalist in Grade 10, and an ICPC Asia Regional Contest silver medalist.

Xinyu Zhang is a 3rd-year undergraduate student at the School of Cyber Science and Engineering, Sichuan University. He has established a good foundation in operating systems, web security, and deep learning. He is an enthusiastic programmer proficient in both front-end and back-end development.



Jiacheng Wu is a freshman at the School of Business, Sichuan University. He is interested in distributed computing. In his free time, he enjoys reading books related to this field. He is delighted to participate in this year's ASC competition and hopes to learn a lot from this competition.

Dr. Kai Gao is the mentor of the team. He is currently an associate research scientist at the School of Cyber Science and Engineering, Sichuan University. His main research interests are in building novel networking systems for reliable and high-performance distributed computing, including network verification, network measurement, and transfer scheduling.

3 TECHNICAL PROPOSAL

We have successfully run all the challenges on our HPC platform. Our highlights include

- We managed to run LLaMA 7B and LLaMA 70B **on a cluster of Cambricon servers, using non-Nvidia technologies**.
- We optimized OpenCAEPoro and achieved 72.4% performance improvement in OBJECT TIME.

3.1 HPC System Design

The CPU cluster consists of 4 machines (numbered as Node0 to Node3) connected through a 10 GbE switch. Detailed configurations of the machines are as shown in Table 1.

The MLU cluster is connected with normal 10GBE fiber.

Table 1. CPU Cluster.

Node 0	
CPU	Intel®Xeon®CPU E5-2609 v4 @ 1.70GHz * 2
Memory	32GiB DIMM DDR4 Synchronous 2400 MHz (0.4 ns) * 2
Disk	44.734TB MR9361-8i
Network	Intel®Corporation 82599ES 10-Gigabit Ethernet
OS	Ubuntu 22.04.3 LTS 5.15.0-78-generic
Compiler	Intel®Fortran Compiler and Intel®C++ Compiler 2024
Node 1-3	
CPU	Intel®Xeon®CPU E5-2650 v4 @ 2.20GHz * 2
Memory	32GiB DIMM DDR4 Synchronous 2400 MHz (0.4 ns) * 12
Disk	1199GB MR9361-8i
Network	Intel®Corporation 82599ES 10-Gigabit Ethernet
OS	Ubuntu 22.04.3 LTS 5.15.0-91-generic
Compiler	Intel®Fortran Compiler and Intel®C++ Compiler 2024

Table 2. MLU Cluster.

Node 0-2	
CPU	Hygon C86 7260 24-core Processor
GPU	MLU370-S4 x 4
Memory	256 GB DIMM DDR4 Synchronous Registered (Buffered) 3200 MHz
OS	Centos 7.6
Compiler	CNCC (Cambricon Compiler Collection)
Node 3-10	
CPU	Hygon C86 7260 24-core Processor
GPU	MLU270-S4 x 4
Memory	256 GB DIMM DDR4 Synchronous Registered (Buffered) 3200 MHz
OS	Centos 7.6
Compiler	CNCC (Cambricon Compiler Collection)

3.2 HPL and HPCG

We have successfully run *HPL* and *HPCG* on our cluster. By adjusting the running parameters, we reached the maximum calculation speed of 2.0826×10^3 GFLOPS and 4.2626 Gflops for *HPL* and *HPCG* respectively, with the efficiency of 82.1% and 0.17%.

3.2.1 System setup. We have successfully run HPL and HPCG in a cluster with 3 machines (Node 1-3) as described in Table 1.

3.2.2 Optimization strategy. In order to make platform reach the best performance, we carefully adjusted the running parameters. For example, we modified *HPL.dat*, the content is shown below. By **making the**

product of P and Q equal to the number of cores and extending the execution time, we can make the calculation speed approach the theoretical maximum as much as possible.

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
1            # of problems sizes (N)
196608         Ns
1            # of NBs
196 256          NBs
1            PMAP process mapping (0=Row-,1=Column-major)
1            # of process grids (P x Q)
8            Ps
9            Qs
16.0         threshold
1            # of panel fact
2            PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
4            NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
1            RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
1            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
1            DEPTHs (>=0)
2            SWAP (0=bin-exch,1=long,2=mix)
64           swapping threshold
0            L1 in (0=transposed,1=no-transposed) form
0            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
8            memory alignment in double (> 0)
##### This line (no. 32) is ignored (it serves as a separator). ######
0                               Number of additional problem sizes for PTRANS
1200 10000 30000                values of N
0                               number of additional blocking sizes for PTRANS
40 9 8 13 13 20 16 32 64        values of NB
```

3.2.3 Final results. For HPL, **we reached the float calculation speed of** 2.0826×10^3 **GFLOPS**. According to the formula of float calculation speed per processor: CPU Frequency $\times$ #cores $\times$ calculations per cycle, the maximum value per processor is 422.4GFLOPS. Therefore, we can calculate the efficiency is $\frac{2082.6}{422.4 \times 3 \times 2} = 82.1\%$

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC11C2R4      196608   196     8     9            2432.82             2.0826e+03
HPL_pdgesv() start time Sat Jan 20 09:47:56 2024

HPL_pdgesv() end time   Sat Jan 20 10:28:29 2024

--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   1.06198309e-03 ...... PASSED
================================================================================

Finished      1 tests with the following results:
              1 tests completed and passed residual checks,
```

Table 3. System setup for LLaMa 2 70B.

Node 0,1	
CPU	Hygon C86 7260 24-core Processor
GPU	MLU370-S4 x 4
Memory	256 GB DIMM DDR4 Synchronous Registered (Buffered) 3200 MHz
OS	Centos 7.6
Compiler	CNCC (Cambricon Compiler Collection)
Pytorch	1.13.1 with Cambricon TORCH_MLU
Python	3.10.8

```
             0 tests completed and failed residual checks,
             0 tests skipped because of illegal input values.
-------------------------------------------------------------------------------

End of Tests.
===============================================================================
```

For HPCG, the tested float calculation speed is 4.2626 Gflops. The efficiency is about 0.17%.

```
Final Summary=
Final Summary::HPCG result is VALID with a GFLOP/s rating of=4.2626
Final Summary::HPCG 2.4 rating for historical reasons is=4.3652
Final Summary::Reference version of ComputeDotProduct used=Performance results are most likely
     suboptimal
Final Summary::Reference version of ComputeSPMV used=Performance results are most likely suboptimal
Final Summary::Reference version of ComputeMG used and number of threads greater than 1=Performance
     results are severely suboptimal
Final Summary::Reference version of ComputeWAXPBY used=Performance results are most likely suboptimal
Final Summary::Results are valid but execution time (sec) is=1761.55
Final Summary::Official results execution time (sec) must be at least=1800
```

3.3 Optimization for LLM inference

3.3.1 System setup. We use the official repo on GitHub to build LLaMA 2 70B system. The hardware is supported by High Performance Computing Platform of School of Cyber Science and Engineering, using 2 nodes with each 4 MLU370S4, connected with normal 10GE fiber.

The distributed method is the default **torch.distributed** module. We follow the guide to port the LLaMA 2 70B to our HPC system. Specifically, we replace codes involved CUDA with MLU-style code, and remove codes with unsupported torch API on Cambricon device. The modified code has been placed in `LLM/llama-mlu`.

3.3.2 Profiling and analysis. **Stage I: Load checkpoints**

Using torch.profiler to profile performance on Python level.

We also get a trace.json using this.

```
with torch.profiler.profile(
        activities=[
            torch.profiler.ProfilerActivity.CPU,
```

Table 4. Basic information about MLU370-S4.

Architecture	Cambricon MLUarch03
Memory Type	LPDDR5
Memory	24GB
Supported	FP32, FP16, BF16, INT16, INT8, INT4

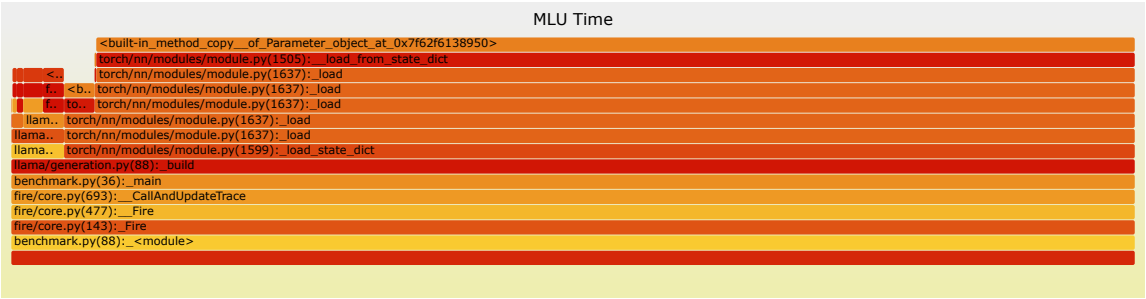


Fig. 2. Flamegraph sorted by MLU Time

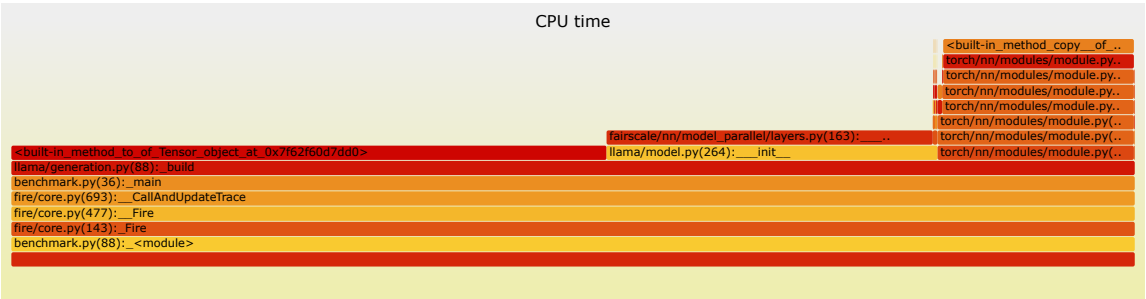


Fig. 3. Flamegraph sorted by CPU Time

```
            torch.profiler.ProfilerActivity.MLU,
        ]
    ) as prof:
        ckpt_path = checkpoints[get_model_parallel_rank()]
        checkpoint = torch.load(ckpt_path, map_location="cpu")
        with open(Path(".") / "params.json", "r") as f:
            params = json.loads(f.read())

        model_args: ModelArgs = ModelArgs(
            max_seq_len=max_seq_len,
            max_batch_size=max_batch_size,
            **params,
        )
        tokenizer = Tokenizer(model_path=tokenizer_path)
        model_args.vocab_size = tokenizer.n_words
        torch.set_default_tensor_type(torch.mlu.HalfTensor)
        model = Transformer(model_args)
```

```
    # checkpoint bf16 to fp16
    for key, value in checkpoint.items():
        if isinstance(value, torch.Tensor):
            checkpoint[key] = checkpoint[key].to(dtype=torch.float16)

    model.load_state_dict(checkpoint, strict=False)
print(prof.key_averages().table(
    sort_by="self_cpu_time_total", row_limit=-1))
prof.export_chrome_trace("./trace-load.json." + str(time.time()))
print(f"Loaded in {time.time() - start_time:.2f} seconds")
```

Stage II: Generate tokens, inference

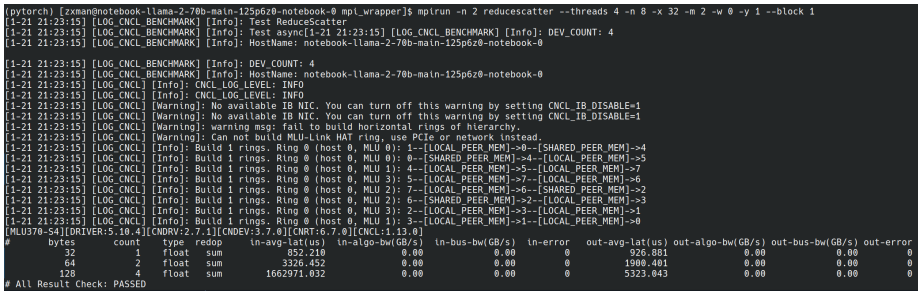
We noticed that during the inference(token generation), the load on each MLU is not balanced. It seems weird because the Python module, fairscale, should have automatically balanced the load. So we assume the fairscale can't connect with MLU driver, so is not supported on MLU. This lead to unflexible `max_batch_size`, since a bigger `max_batch_size` will break the balance between MLU and torch processes, and cause a very slow speed on inference.

The reason we didn't do a torch.profiler is it will break the `torch.set_grad_enabled(False)`, which is should be `torch.inference_mode()` but not supported by MLU-pytorch. Here's the reference.

3.3.3 *Optimization strategy.* Connection: CNCL Perf and Optimize

Cambricon provided a customized Linux perf named CNPerf, and when we monitor the PCIe speed is extremely slow (about 20MB per second).

And we tested it, the CNCL Library (like NCCL on MLU) have an amazing bandwidth and latency so we sure there must be issues in the distributed inference.



```
(pytorch) [zxman@notebook-llama-2-70b-main-125p6z0-notebook-0 mpi(_wrapper]$ mpirun -n 2 reducescatter --threads 4 -n 8 -x 32 -m 2 -w 0 -y 1 --block 1
[1-21 21:23:15] [LOG_CNCL_BENCHMARK] [Info]: Test ReduceScatter
[1-21 21:23:15] [LOG_CNCL_BENCHMARK] [Info]: Test async[1-21 21:23:15] [LOG_CNCL_BENCHMARK] [Info]: DEV_COUNT: 4
[1-21 21:23:15] [LOG_CNCL_BENCHMARK] [Info]: HostName: notebook-llama-2-70b-main-125p6z0-notebook-0

[1-21 21:23:15] [LOG_CNCL_BENCHMARK] [Info]: DEV_COUNT: 4
[1-21 21:23:15] [LOG_CNCL_BENCHMARK] [Info]: HostName: notebook-llama-2-70b-main-125p6z0-notebook-0
[1-21 21:23:15] [LOG_CNCL] [Info]: CNCL_LOG_LEVEL: INFO
[1-21 21:23:15] [LOG_CNCL] [Info]: CNCL_LOG_LEVEL: INFO
[1-21 21:23:15] [LOG_CNCL] [Warning]: No available IB NIC. You can turn off this warning by setting CNCL_IB_DISABLE=1
[1-21 21:23:15] [LOG_CNCL] [Warning]: No available IB NIC. You can turn off this warning by setting CNCL_IB_DISABLE=1
[1-21 21:23:15] [LOG_CNCL] [Warning]: warning msg: fail to build horizontal rings of hierarchy.
[1-21 21:23:15] [LOG_CNCL] [Warning]: Can not build MLU-Link HAT ring, use PCIe or network instead.
[1-21 21:23:15] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 0): 1--[LOCAL_PEER_MEM]->0--[SHARED_PEER_MEM]->4
[1-21 21:23:15] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 0): 0--[SHARED_PEER_MEM]->4--[LOCAL_PEER_MEM]->5
[1-21 21:23:15] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 1): 4--[LOCAL_PEER_MEM]->5--[LOCAL_PEER_MEM]->7
[1-21 21:23:15] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 3): 5--[LOCAL_PEER_MEM]->7--[LOCAL_PEER_MEM]->6
[1-21 21:23:15] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 2): 7--[LOCAL_PEER_MEM]->6--[SHARED_PEER_MEM]->2
[1-21 21:23:15] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 2): 6--[SHARED_PEER_MEM]->2--[LOCAL_PEER_MEM]->3
[1-21 21:23:15] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 3): 2--[LOCAL_PEER_MEM]->3--[LOCAL_PEER_MEM]->1
[1-21 21:23:15] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 1): 3--[LOCAL_PEER_MEM]->1--[LOCAL_PEER_MEM]->0
[MLU370-S4][DRIVER:5.10.4][CNDRV:2.7.1][CNDEV:3.7.0][CNRT:6.7.0][CNCL:1.13.0]
#       bytes       count    type  redop     in-avg-lat(us)  in-algo-bw(GB/s)  in-bus-bw(GB/s)  in-error   out-avg-lat(us) out-algo-bw(GB/s) out-bus-bw(GB/s) out-error
          32           1    float   sum            852.210              0.00              0.00         0           926.881              0.00              0.00         0
          64           2    float   sum           3326.452              0.00              0.00         0          1900.401              0.00              0.00         0
         128           4    float   sum        1662971.032              0.00              0.00         0          5323.043              0.00              0.00         0
# All Result Check: PASSED
```

Fig. 4. CNCL Benchmark Result

3.3.4 *Final results.*

We didn't use the framework by hugging face but programmed one. Since we started this task very late, we just run it on our HPC system and do some profiler work, having limited time to solve it. So it inference really slow and due to the `max_batch_size` issue the prompt can't be too long. Here's the result:

```
> initializing model parallel with size 8
> initializing ddp with size 1
```

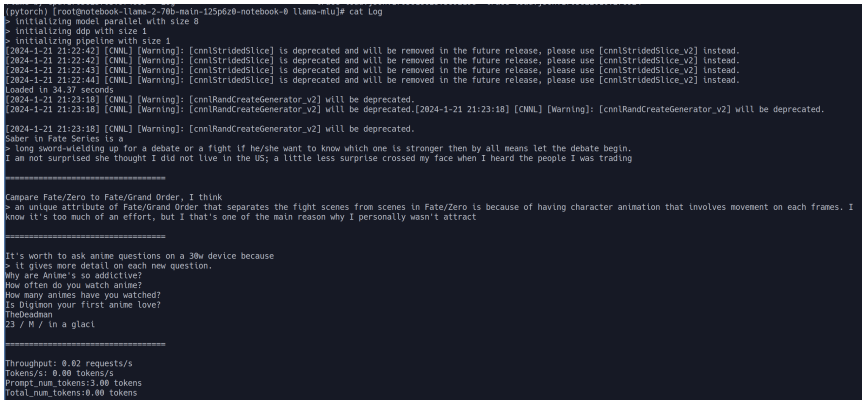


Fig. 5. Enter Caption

```
> initializing pipeline with size 1
[2024-1-21 21:22:42] [CNNL] [Warning]: [cnnlStridedSlice] is deprecated and will be removed in the
     future release, please use [cnnlStridedSlice_v2] instead.
[2024-1-21 21:22:42] [CNNL] [Warning]: [cnnlStridedSlice] is deprecated and will be removed in the
     future release, please use [cnnlStridedSlice_v2] instead.
[2024-1-21 21:22:43] [CNNL] [Warning]: [cnnlStridedSlice] is deprecated and will be removed in the
     future release, please use [cnnlStridedSlice_v2] instead.
[2024-1-21 21:22:44] [CNNL] [Warning]: [cnnlStridedSlice] is deprecated and will be removed in the
     future release, please use [cnnlStridedSlice_v2] instead.
Loaded in 34.37 seconds
[2024-1-21 21:23:18] [CNNL] [Warning]: [cnnlRandCreateGenerator_v2] will be deprecated.
[2024-1-21 21:23:18] [CNNL] [Warning]: [cnnlRandCreateGenerator_v2] will be deprecated.[2024-1-21
     21:23:18] [CNNL] [Warning]: [cnnlRandCreateGenerator_v2] will be deprecated.

[2024-1-21 21:23:18] [CNNL] [Warning]: [cnnlRandCreateGenerator_v2] will be deprecated.
Saber in Fate Series is a
> long sword-wielding up for a debate or a fight if he/she want to know which one is stronger then by
     all means let the debate begin.
I am not surprised she thought I did not live in the US; a little less surprise crossed my face when I
      heard the people I was trading

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

Campare Fate/Zero to Fate/Grand Order, I think
> an unique attribute of Fate/Grand Order that separates the fight scenes from scenes in Fate/Zero is
     because of having character animation that involves movement on each frames. I know it's too much
      of an effort, but I that's one of the main reason why I personally wasn't attract

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

It's worth to ask anime questions on a 30w device because
> it gives more detail on each new question.
Why are Anime's so addictive?
How often do you watch anime?
How many animes have you watched?
Is Digimon your first anime love?
TheDeadman
23 / M / in a glaci

==================================
```

```
Throughput: 0.02 requests/s
Tokens/s: 0.00 tokens/s
Prompt_num_tokens:3.00 tokens
Total_num_tokens:0.00 tokens



[I debug.cpp:49] [c10d] The debug level is set to DETAIL.
WARNING:torch.distributed.run:
*****************************************
Setting OMP_NUM_THREADS environment variable for each process to be 1 in default, to avoid your system
      being overloaded, please further tune the variable for optimal performance in your application
     as needed.
*****************************************
[I socket.cpp:442] [c10d - debug] The server socket will attempt to listen on an IPv6 address.
[I socket.cpp:492] [c10d - debug] The server socket is attempting to listen on [::]:25900.
[I socket.cpp:566] [c10d] The server socket has started to listen on [::]:25900.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
     (10.233.94.63, 25900).
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
     main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
     .cluster.local]:25900.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
     notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
     local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
     ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57196.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
     notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
     bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57196.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
     notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
     bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38678.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
     notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
     bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38680.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
     (10.233.94.63, 25900).
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
     main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
     .cluster.local]:25900.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
     notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
     bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57200.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
     notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
     local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
     ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57200.
[I debug.cpp:49] [c10d] The debug level is set to DETAIL.
[I debug.cpp:49] [c10d] The debug level is set to DETAIL.
[I debug.cpp:49] [c10d] The debug level is set to DETAIL.
[I debug.cpp:49] [c10d] The debug level is set to DETAIL.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
     (10.233.94.63, 25900).
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
     main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
     .cluster.local]:25900.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
     notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
     local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
     ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57212.
```

Proposal for Asia Super Computing (ASC) 2024

```
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57212.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
    (10.233.94.63, 25900).
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
    main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
    .cluster.local]:25900.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
    notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
    local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
    ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57214.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57214.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38682.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38684.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38686.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38688.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38690.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38692.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
    (10.233.94.63, 25900).
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
    main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
    .cluster.local]:25900.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
    (10.233.94.63, 25900).
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57216.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
    notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
    local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
    ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57216.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
    (10.233.94.63, 25900).
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
    main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
    .cluster.local]:25900.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57218.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
    notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
    local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
    ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57218.
```

```
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
    main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
    .cluster.local]:25900.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
    notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
    local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
    ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57220.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57220.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
    (10.233.94.63, 25900).
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
    main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
    .cluster.local]:25900.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57222.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
    notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
    local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
    ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57222.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
    (10.233.94.63, 25900).
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
    main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
    .cluster.local]:25900.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
    notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
    local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
    ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57224.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57224.
[I socket.cpp:627] [c10d - debug] The client socket will attempt to connect to an IPv6 address of
    (10.233.94.63, 25900).
[I socket.cpp:699] [c10d - trace] The client socket is attempting to connect to [notebook-llama-2-70b-
    main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc
    .cluster.local]:25900.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57226.
[I socket.cpp:787] [c10d] The client socket has connected to [notebook-llama-2-70b-main-125p6z0-
    notebook-0.notebook-llama-2-70b-main-125p6z0.ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.
    local]:25900 on [notebook-llama-2-70b-main-125p6z0-notebook-0.notebook-llama-2-70b-main-125p6z0.
    ns-bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:57226.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38694.
[I socket.cpp:298] [c10d - debug] The server socket on [::]:25900 has accepted a connection from [
    notebook-llama-2-70b-main-125p6z0-worker-0.notebook-llama-2-70b-main-125p6z0.ns-
    bc0346a826e64e0e93036c7e8e0e240a.svc.cluster.local]:38696.
/root/algorithm/llama-mlu/llama/generation.py:139: UserWarning:  MLU operators don't support 64-bit
    calculation. so the 64 bit data will be forcibly converted to 32-bit for calculation.  (Triggered
     internally at /torch/catch/torch_mlu/csrc/aten/utils/tensor_util.cpp:159.)
tokens[k, : len(t)] = torch.tensor(t, dtype=torch.long, device="mlu")
/root/algorithm/llama-mlu/llama/generation.py:139: UserWarning:  MLU operators don't support 64-bit
    calculation. so the 64 bit data will be forcibly converted to 32-bit for calculation.  (Triggered
     internally at /torch/catch/torch_mlu/csrc/aten/utils/tensor_util.cpp:159.)
tokens[k, : len(t)] = torch.tensor(t, dtype=torch.long, device="mlu")
[1-21 21:23:3] [LOG_CNCL] [Info]: CNCL_LOG_LEVEL: INFO
```

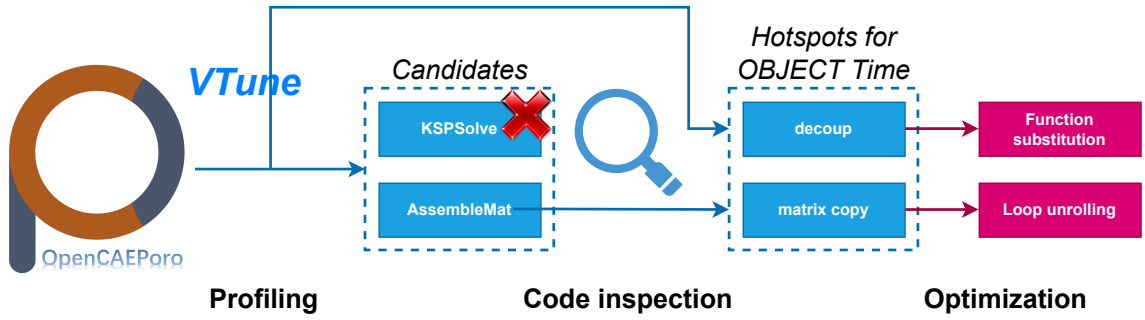


Fig. 6. The analysis process for OpenCAEPoro.

```
[1-21 21:23:3] [LOG_CNCL] [Info]: CNCL_LOG_LEVEL: INFO
/root/algorithm/llama-mlu/llama/generation.py:139: UserWarning:  MLU operators don't support 64-bit
     calculation. so the 64 bit data will be forcibly converted to 32-bit for calculation.  (Triggered
      internally at /torch/catch/torch_mlu/csrc/aten/utils/tensor_util.cpp:159.)
tokens[k, : len(t)] = torch.tensor(t, dtype=torch.long, device="mlu")
[1-21 21:23:6] [LOG_CNCL] [Info]: CNCL_LOG_LEVEL: INFO
/root/algorithm/llama-mlu/llama/generation.py:139: UserWarning:  MLU operators don't support 64-bit
     calculation. so the 64 bit data will be forcibly converted to 32-bit for calculation.  (Triggered
      internally at /torch/catch/torch_mlu/csrc/aten/utils/tensor_util.cpp:159.)
tokens[k, : len(t)] = torch.tensor(t, dtype=torch.long, device="mlu")
[1-21 21:23:17] [LOG_CNCL] [Info]: CNCL_LOG_LEVEL: INFO
[1-21 21:23:17] [LOG_CNCL] [Warning]: warning msg: fail to build horizontal rings of hierarchy.
[1-21 21:23:17] [LOG_CNCL] [Warning]: Can not build MLU-Link HAT ring, use PCIe or network instead.
[1-21 21:23:17] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 0): 7--[SOCKET]->0--[
     SHARED_PEER_MEM]->1
[1-21 21:23:17] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 1): 0--[SHARED_PEER_MEM]->1--[
     SHARED_PEER_MEM]->2
[1-21 21:23:17] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 2): 1--[SHARED_PEER_MEM]->2--[
     SHARED_PEER_MEM]->3
[1-21 21:23:17] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 0, MLU 3): 2--[SHARED_PEER_MEM]->3--[
     SOCKET]->4
[1-21 21:23:17] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 1, MLU 0): 3--[SOCKET]->4--[
     SHARED_PEER_MEM]->5
[1-21 21:23:17] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 1, MLU 1): 4--[SHARED_PEER_MEM]->5--[
     SHARED_PEER_MEM]->6
[1-21 21:23:17] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 1, MLU 2): 5--[SHARED_PEER_MEM]->6--[
     SHARED_PEER_MEM]->7
[1-21 21:23:17] [LOG_CNCL] [Info]: Build 1 rings. Ring 0 (host 1, MLU 3): 6--[SHARED_PEER_MEM]->7--[
     SOCKET]->0
[1-21 21:25:32] [LOG_CNCL] [Info]: Abort communication tasks on comm0
[1-21 21:25:32] [LOG_CNCL] [Info]: Abort communication tasks on comm2
[1-21 21:25:32] [LOG_CNCL] [Info]: Abort communication tasks on comm3
[1-21 21:25:32] [LOG_CNCL] [Info]: Abort communication tasks on comm1
```

Original Logs (both master and slave node) are placed in the `LLM/Log`

3.4 The OpenCAEPoro Challenge

In this challenge, we set up the environment on a cluster containing 4 machines with **Intel compilers** and NFS. We identify two hotspots and optimize the program with **function substitution** and **loop unrolling**

Listing 1. Shell script to configure environment variables for OpenCAEPoro.

```
export INTEL_ENV_PROMPT="(intel)␣"
if [ ! $(echo "$PS1" | grep -q "$INTEL_ENV_PROMPT") ]; then
    export PS1="${INTEL_ENV_PROMPT}${PS1:-}"
fi

export OCAEP_ROOT='/home/asc/cloud/OpenCAEPoro_ASC2024'

export ASC_CC='mpiicx'
export ASC_CXX='mpiicpx'
export ASC_FC='mpiifx'

export ASC_CORE=16

#-----------------
export CC=$ASC_CC
export CXX=$ASC_CXX
export FC=$ASC_FC

. /home/asc/intel/oneapi/2024.0/oneapi-vars.sh --force
```

respectively, as shown in Figure 6. With the two optimizations combined, we achieve **72.4% of performance improvement in OBJECT TIME (58.013s)**, and 16.1% improvement of the overall simulation time.

3.4.1 System setup. In this section, we introduce our basic hardware/software environment and additional efforts to build and run the OpenCAEPoro software.

Basic hardware and software setup. We use the CPU cluster (as reported in Table 1) to build and run OpenCAEPoro.

Building OpenCAEPoro. We use Intel compilers and Intel MPI libraries for compilation and execution. Compiling OpenCAEPoro on our platform requires additional system configurations, and we have developed a shell script to simplify the process, as shown in Listing 1.

We also modify the compilation scripts in the OpenCAEPoro code base to properly configure the compiler used to build the components of OpenCAEPoro. There are also a few compilation problems that we encounter:

- We find that after switching the compiler, *petsc_solver* is not recompiled and causes an error. Thus, we must manually remove the old build path with `rm -rf build` when using *cmake 3.22.1*, environment variables do not take effect. Supplementary definitions are shown below:

  ```
  set(PETSC_DIR "$ENV{OCAEP_ROOT}/petsc-3.19.3/")
  ```

  However, if we only modify the code and do not switch the compiler, we must skip the `rm -rf build` command to leverage incremental compilation.
- If the system contains *parmetis*, linking the library in the system will cause *MPI* errors. Thus, we must explicitly set the variables as below.

  ```
  set(PARMETIS_DIR $ENV{PARMETIS_DIR})
  set(METIS_DIR $ENV{METIS_DIR})
  ```

Configuring NFS. We use the following NFS configuration:

```
/home/asc/cloud *(rw,sync,no_root_squash,no_subtree_check)
```

Running OpenCAEPoro. After OpenCAEPoro is properly compiled and NFS properly running, we use the following command to run OpenCAEPoro on our cluster:

```
mpirun --hostfile myhost ./testOpenCAEPoro ./data/case1/case1.data verbose=1
```

The content of the "myhost" file is as shown follows:

```
10.24.1.4:24
10.24.1.5:24
10.24.1.6:24
10.24.1.3:16
```

3.4.2 Performance profiling and analysis. To identify the potential performance bottlenecks, we use profiling tools and inspect the source code, as shown in Figure 6. We find that this program is sufficiently optimized by using multiple threads. The functions which we can try to optimize are **FIM_Solver_p_cpr** and **OCPFlux01::AssembleMatFIM**. The **dgemm_** function is also worth noting if we have special hardware to accelerate the calculation.

Profiling with Intel VTune. To analyze the performance of *OpenCAEPoro* program, we use *Intel Vtune Profiler*, which is designed for Intel x86 processors.

(1) Install the latest version of "VTune Profiler" on this machine and configure password-free login to the server. To explicitly analyze the program invoked by *mpirun*, the analyzed configuration screenshot is shown in Figure 7 (in particular pay attention to the "Trace MPI" option):
(2) Select hotspots on the right of Figure 7, and after waiting for an hour, the analysis results are as shown in Figure 8: It can be found that the **CPU utilization is relatively ideal**.
(3) Check the corresponding flame graph. It can be seen that the two functions **KSPSolve** and **AssembleMat** take more time, as in Figure 9.

Code inspection. We first inspect the definition of OBJECT TIME in the source code as below (the source code is reformatted for better readability):

```
cout << endl << "OBJECT␣TIME:␣" << setw(fixWidth)
     << OCPTIME_TOTAL + MATRIX_CONVERSION_TIME - OCPTIME_LSOLVER << "␣s␣"
     << endl << endl;
```

As **OCPTIME_LSOLVER** is subtracted from the overall time, we conclude that **the performance of KSPSolve is not the bottleneck for OBJECT TIME**. Thus, we focus on **AssembleMat** and further inspect the Top-down Tree, as shown in Figure 10.

According to the previous analysis, **VectorPetscSolver::Solve** cannot be modified. At the same time, **DaABpbC** calls another library function **dgemm_**, which is also a built-in function specifically optimized for matrix multiplication and cannot be modified. Therefore, the functions that we should really

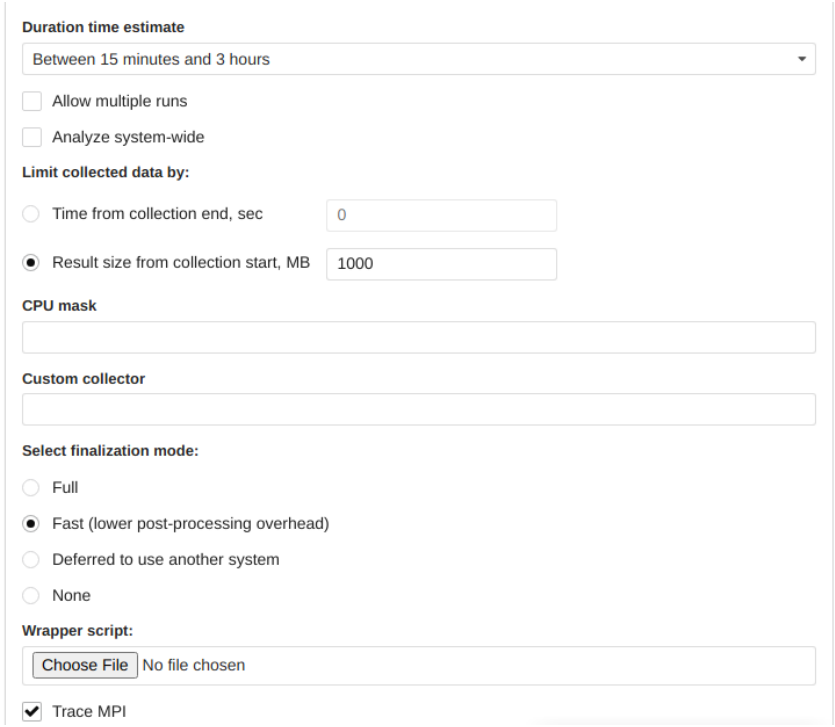


Fig. 7. VTune Profiler advanced configuration for MPI program

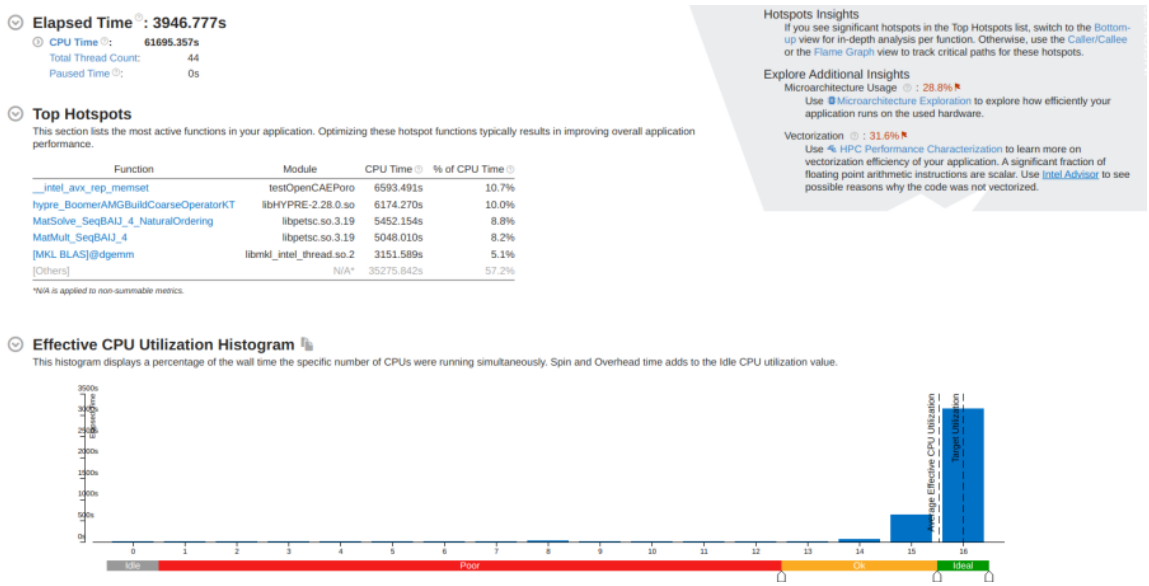


Fig. 8. VTune Profiler CPU analysis result

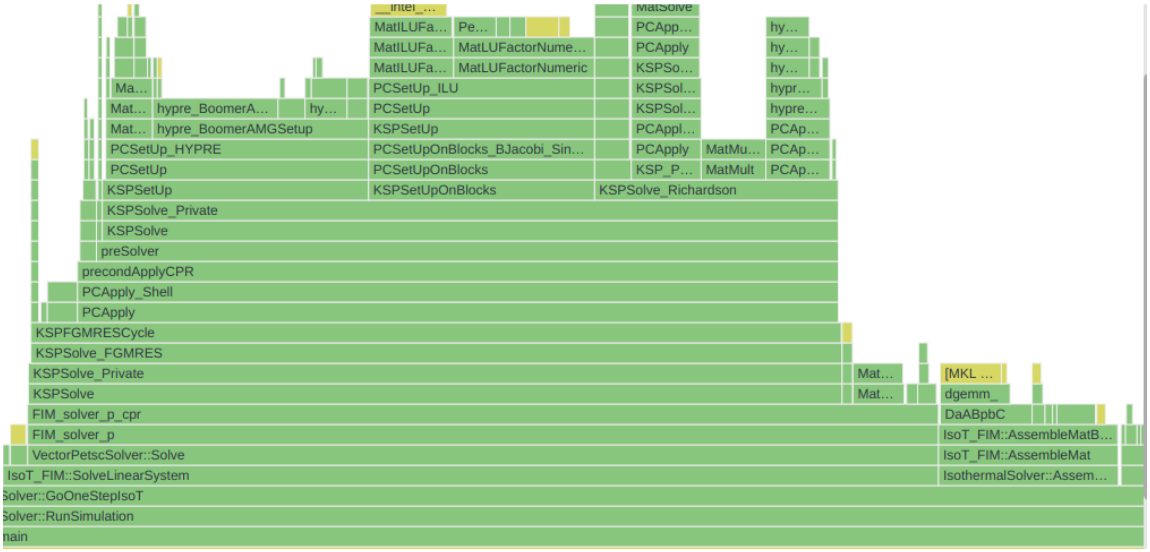


Fig. 9. VTune Profiler CPU analysis flame graph

Source Function Stack	CPU Time: Total ▼	CPU Time: Self	Function (Full)
Total	100.0%	0s	
_start	100.0%	0s	_start
__libc_start_main	100.0%	0s	__libc_start_main
main	100.0%	0s	main
Solver::RunSimulation	99.9%	0s	Solver::RunSimulation(Reservoir&, OCPContr…
Solver::GoOneStepIsoT	99.9%	0s	Solver::GoOneStepIsoT(Reservoir&, OCPCont…
IsoT_FIM::SolveLinearSystem	80.6%	0s	IsoT_FIM::SolveLinearSystem(LinearSystem&,…
VectorPetscSolver::Solve	78.5%	0s	VectorPetscSolver::Solve(void)
PetscSolver::AssembleMat	1.5%	82.944s	PetscSolver::AssembleMat(std::vector<std::ve…
IsoT_FIM::GetSolution	0.6%	355.497s	IsoT_FIM::GetSolution(Reservoir&, std::vector…
LinearSystem::ClearData	0.1%	30.110s	LinearSystem::ClearData(void)
PetscSolver::CalCommTerm	0.0%	0s	PetscSolver::CalCommTerm(unsigned int cons…
std::chrono::_V2::steady_clock::now	0.0%	0.020s	std::chrono::_V2::steady_clock::now(void)
OCPNRsuite::UpdateIter	0.0%	0.012s	OCPNRsuite::UpdateIter(unsigned int const&)
AllWells::GetNumOpenWell	0.0%	0.008s	AllWells::GetNumOpenWell(void) const
IsothermalSolver::AssembleMat	15.5%	0s	IsothermalSolver::AssembleMat(Reservoir con…
IsoT_FIM::AssembleMat	15.5%	0.008s	IsoT_FIM::AssembleMat(LinearSystem&, Res…
IsoT_FIM::AssembleMatBulks	15.4%	657.349s	IsoT_FIM::AssembleMatBulks(LinearSystem&,…
DaABpbC	7.9%	1169.699s	DaABpbC(int const&, int const&, int const&, do…
OCPFlux01::AssembleMatFIM	3.3%	1921.960s	OCPFlux01::AssembleMatFIM(BulkConnPair c…
Dscalar	1.1%	114.116s	Dscalar(int const&, double const&, double*)
std::vector<double, std::allocator<double>>	0.7%	454.027s	std::vector<double, std::allocator<double>>::in…

Fig. 10. VTune Profiler CPU analysis top-down tree

care about include other functions in the modifiable part in `PETSC_FIM_solver.cpp`, in particular the `FIM_Solver_p_cpr` function, and the contents of the function `OCPFlux01::AssembleMatFIM`.

3.4.3 Optimizations. Based on further investigations, we identify two functions that are hotspots of the modifiable parts. First, we find that the functions in `decoup.cpp`, especially the `decoup_abf_4x4` function,

are invoked very frequently. To reduce the cost of calling time, we first check if the variable `nb` is equal to 4. It turned out to be true.

Therefore, we expand the `decoup_abf_4x4` as a fast path when `nb` is equal to 4. The expanded code segment is as shown below:

```
start = MPI_Wtime();
int nrow = nBlockRows;
if (nb != 4) decoup(val, rhs, rpt, cpt, nb, nBlockRows, DecoupType, is_thermal);
else {
    // expanding decoup_abf_4x4 here
    double *tmpval = val, *tmprhs = rhs;

    double *smat = (double *)malloc(nb2 * sizeof(double));
    double *binv = (double *)malloc(nb2 * sizeof(double));
    double *svec = (double *)malloc(nb * sizeof(double));

    int j, ibegin, iend;
    for (i = 0; i < nrow; ++i)
    {
        memcpy(binv, tmpval, nb2 * sizeof(double));
        smat_inv_4x4(binv);
        ibegin = rpt[i];
        iend = rpt[i + 1];
        for (j = ibegin; j < iend; ++j)
        {
            if (j != ibegin)
            {
                smat_mul_4x4(binv, tmpval, smat);
                memcpy(tmpval, smat, nb2 * sizeof(double));
            }
            else
            {
                smat_identity_4x4(tmpval);
            }
            tmpval += nb2;
        }
        smat_vec_mul_4(binv, tmprhs, smat);
        memcpy(tmprhs, smat, nb * sizeof(double));
        tmprhs += nb;
    }

    free(smat);
    free(binv);
    free(svec);
}
```

Second, we find that there are a lot of loops in the modifiable part of `PETSC_FIM_solver.cpp`. For example, Lines 193-233 use loops to compute the vectors of a fixed size, `globalx` and `globaly`. By unrolling the loops, the code segment can leverage *single-instruction multi-data parallelism*. See the change of the example code segment as shown in Figure 11.

We also tried to optimize the inverse of 4x4 matrix function, but it did little help and didn't affect the OBJECT TIME obviously.

```
// source code before the change
for (Ii = 0; Ii < nBlockRows; Ii++)
{
  for (i = 0; i < blockSize; i++)
  {
    globalx[i] = (Ii + Istart) * blockSize + i;
  }

  for (int i = rpt[Ii]; i < rpt[Ii + 1]; i++)
  {
    for (int j = 0; j < blockSize; j++)
    {
      globaly[j] = cpt[i] * blockSize + j;
    }
    ierr = MatSetValues(A, blockSize, global,
                        blockSize, globally,
                        valpt, INSERT_VALUES);
    CHKERRQ(ierr);
    valpt += blockSize * blockSize;
  }
}
```

```
// Source code after unrolling
for (Ii = 0; Ii < nBlockRows; Ii++)
{
  globalx[0] = ((Ii + Istart) << 2);
  globalx[1] = ((Ii + Istart) << 2) + 1;
  globalx[2] = ((Ii + Istart) << 2) + 2;
  globalx[3] = ((Ii + Istart) << 2) + 3;

  for (int i = rpt[Ii]; i < rpt[Ii + 1]; i++)
  {
    globaly[0] = (cpt[i] << 2);
    globaly[1] = (cpt[i] << 2) + 1;
    globaly[2] = (cpt[i] << 2) + 2;
    globaly[3] = (cpt[i] << 2) + 3;
    ierr = MatSetValues(A, 4, globalx,
                        4, globally,
                        valpt, INSERT_VALUES);
    CHKERRQ(ierr);
    valpt += 16;
  }
}
```

Fig. 11. Example of loop unrolling in OpenCAEPoro.

3.4.4 Final results. After applying these algorithms, we reduced the OBJECT time from around **100 seconds** to around **58 seconds**. We can also observe that there is an improvement in **Linear Solver** (from 1750 seconds to 1566 seconds, 10.5% of improvement), which is out of scope of our modifiable area. The overall simulation time improvement is 16.1%, from 1817 seconds to 1566 seconds.

```
==================================================
Item                          Average    Variance         Max         Min
Grid Num                         4000         105        4125        3496
Updating Properties(s)          9.748       3.135      16.007       7.209
Assembling(s)                  28.557       4.834      38.703      22.786
Communication(collect)(s)      23.548      10.815      33.836       2.326
Communication(P2P)(s)           0.747       0.309       1.588       0.259
1AllReduce(OCPCheck)(s)         2.417       0.907       3.421       0.638
Newton Step(s)                  1.819       0.639       3.005       0.980
Newton Step(c)(s)               1.304       0.543       2.404       0.830
==================================================
Final time:                   3600.000( Day )
 - Avg time step size .......   10.714 (336 steps)
 - Avg Newton steps .........    3.914 (1315 succeeded + 101 wasted)
 - Avg linear steps .........    6.341 (8338 succeeded + 1779 wasted)
Simulation time:                1817.387 (Seconds)
 - % Input & Partition .........    0.013 (0.235s)
 - % Partition - ParMetis ......    0.006 (0.101s)
 - % Input Reservoir ...........    0.004 (0.074s)
 - % Setup Simulator ...........    0.000 (0.004s)
 - % Initialization ............    0.001 (0.021s)
 - % Assembling ................    1.349 (24.514s)
 - % Assembling for LS interface    0.960 (17.440s)
 - % Linear Solver .............   96.313 (1750.376s)
 - % Newton Step ................    0.062 (1.125s)
 - % Updating Properties .......    0.451 (8.193s)
 - % Output ....................    0.027 (0.492s)
==================================================
```

```
OBJECT TIME:    99.981 s

==================================================
Item                          Average     Varance         Max         Min
Grid Num                         5333         211        5498        4303
Updating Properties(s)          7.810       0.332       8.508       6.925
Assembling(s)                  35.873       1.814      39.665      30.980
Communication(collect)(s)       6.640       2.330      13.266       2.605
Communication(P2P)(s)           0.528       0.096       0.766       0.297
1AllReduce(OCPCheck)(s)         0.843       0.185       1.145       0.416
Newton Step(s)                  1.621       0.165       2.059       1.339
Newton Step(c)(s)               1.320       0.123       1.586       1.031
==================================================
Final time:                   3600.000( Day )
 - Avg time step size .......   10.315 (349 steps)
 - Avg Newton steps .........    3.871 (1351 succeeded + 145 wasted)
 - Avg linear steps .........    6.260 (8457 succeeded + 2601 wasted)
Simulation time:                1566.801 (Seconds)
 - % Input & Partition .........    0.015 (0.234s)
 - % Partition - ParMetis ......    0.006 (0.091s)
 - % Input Reservoir ...........    0.005 (0.073s)
 - % Setup Simulator ...........    0.000 (0.005s)
 - % Initialization ............    0.002 (0.024s)
 - % Assembling ................    2.420 (37.920s)
 - % Assembling for LS interface    0.392 (6.136s)
 - % Linear Solver .............   96.324 (1509.212s)
 - % Newton Step ...............    0.092 (1.440s)
 - % Updating Properties .......    0.516 (8.088s)
 - % Output ...................    0.027 (0.423s)
==================================================

OBJECT TIME:    58.013 s
```