zoya.masih@gdwg.de

Zoya Masih

Trace Analysis of ML workflows using Vampir

Table of contents

Table of Contents

What is Vampir

- Vampir is a framework for performance analysis
- Enables developers to study program behavior at a fine level of detail
- The tool is well-proven and widely used in HPC.

Vampir reads OTF2 Files

- Vampir reads Open Trace Format 2 (OTF2) files
- OTF2 is specially designed for massively parallel programs
- Some performance monitoring environments produce OTF2 trace files
  - like Score-P and TAU

Score-P: The main instrumentor

- compile and run a program, with Score-P prefixed

```
scorep gcc -o myApp [-fopenmp] myApp.c
scorep ./myApp
>> output directory: scorep-20240331_1326_33868353060433708
```

Score-P: The main instrumentor

- Enabling the tracing mode generates an OTF2 file
- export SCOREP_ENABLE_TRACING=true
- This file is used to capture and store performance data

Table of Contents

The Python-binding

- Score-P only supports Fortran and C/C++ applications by default
- Score-P Python module, supports Python code as well
- The Score-P C-bindings serve as a bridge between Python, C, and Score-P
- Facilitating communication and interaction between these components

Python-binding

Listing: Installing python-binding

```
export PATH=/path/to/scorep/bin:\$PATH

 git clone https://github.com/score-p/scorep\_binding\_python
 cd scorep_python_bindings/
 pip3 install .
 cd ..

 python -m scorep <script.py>
```

Python-binding

Listing: Using Score-P for python-binding

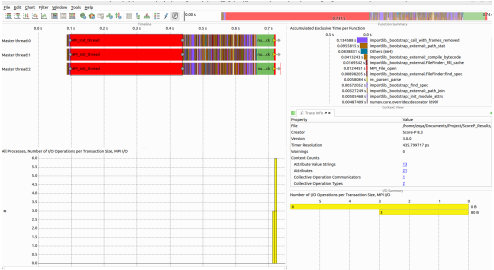
```
export SCOREP_ENABLE_TRACING=true
export SCOREP_TOTAL_MEMORY=1837MB #estimate with scorep-score
python -m scorep --io=posix [--mpp=mpi]  --thread=pthread script.py
srun -n 2 python -m scorep script.py
```

Table of Contents

A simple MPI4PY code

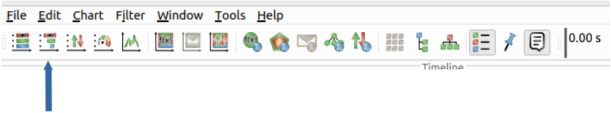
- `srun -n 3 python -m scorep  --io=posix --mpp=mpi script.py`

```
from mpi4py import MPI
import numpy as np
amode =
    ↪ MPI.MODE_WRONLY|MPI.MODE_CREATE
comm = MPI.COMM_WORLD
fh = MPI.File.Open(comm,
    ↪ "./datafile.contig", amode)
buffer = np.empty(10, dtype=np.int)
buffer[:] = comm.Get_rank()
offset =
    ↪ comm.Get_rank()*buffer.nbytes
fh.Write_at_all(offset, buffer)
fh.Close()
```

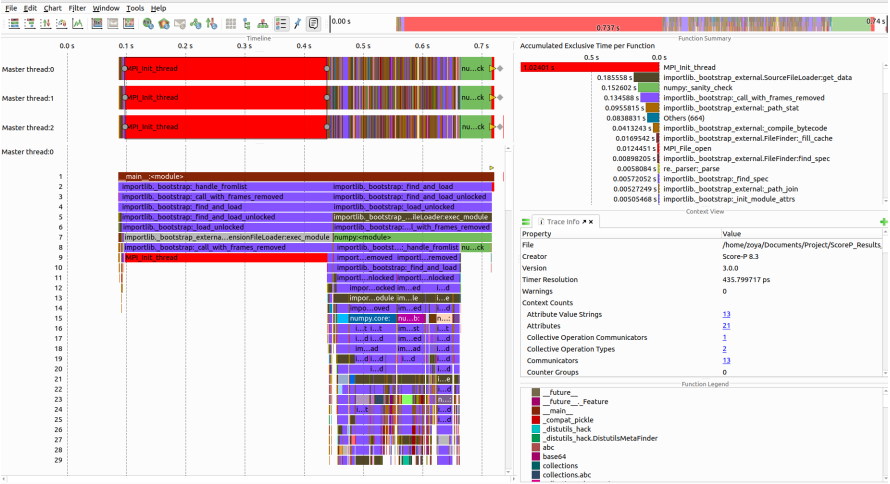


Process Timeline

- Information about different Levels of function calls
- levels show the execution flow and hierarchical relationships

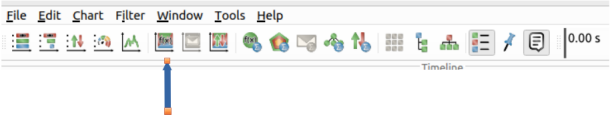


Process Timeline

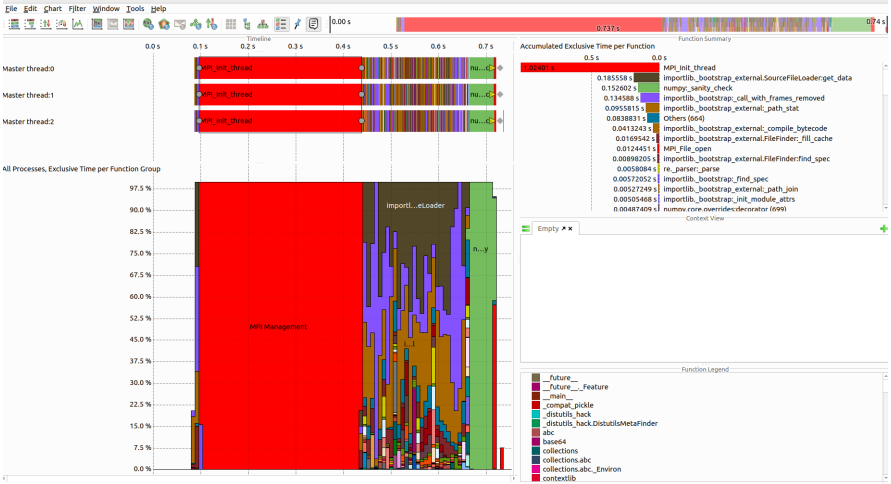


Summary Timeline

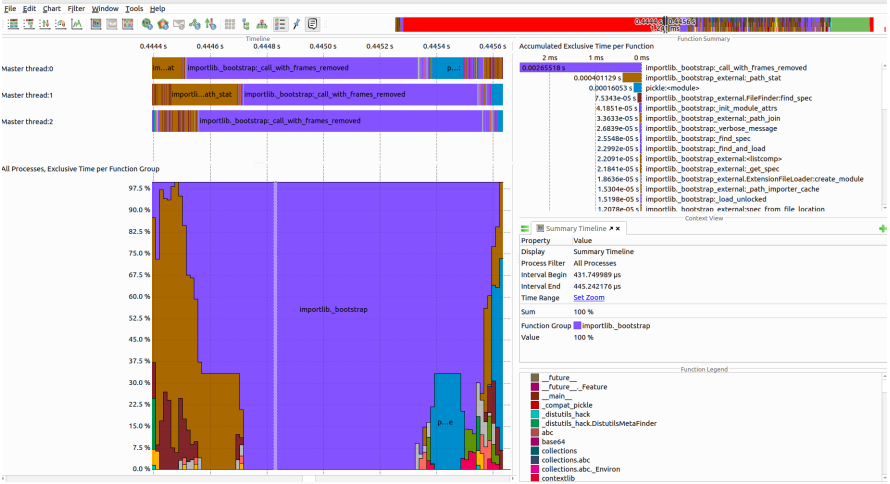
- Provides detailed information about functions, communications, and synchronization



Summary Timeline

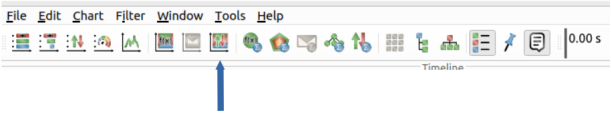


Summary Timeline

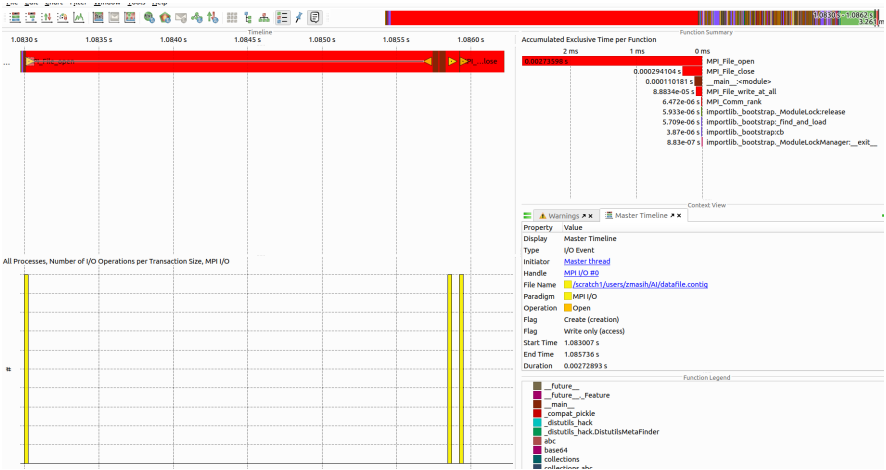


I/O summary

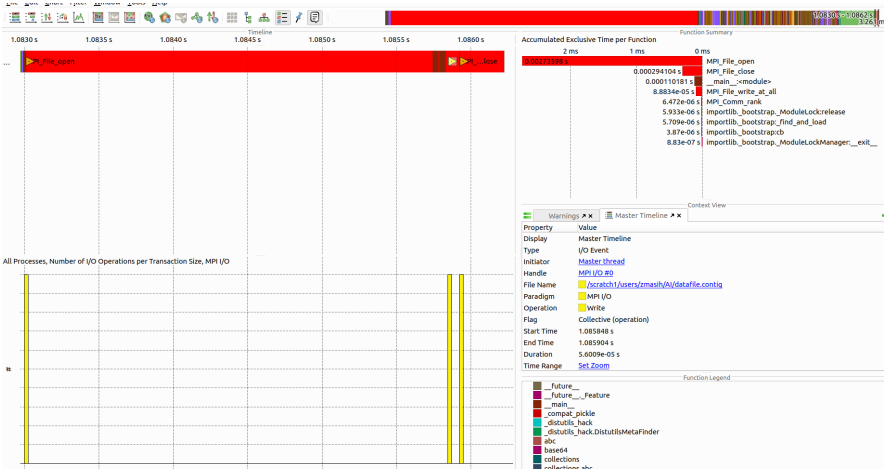
- Vampir highlights I/O operations if I/O performance data has been recorded.
  - –io=posix (=runtime:posix)
- MPI I/O routines are automatically instrumented
  - if the scorep instrumenter detects the MPI programming paradigm



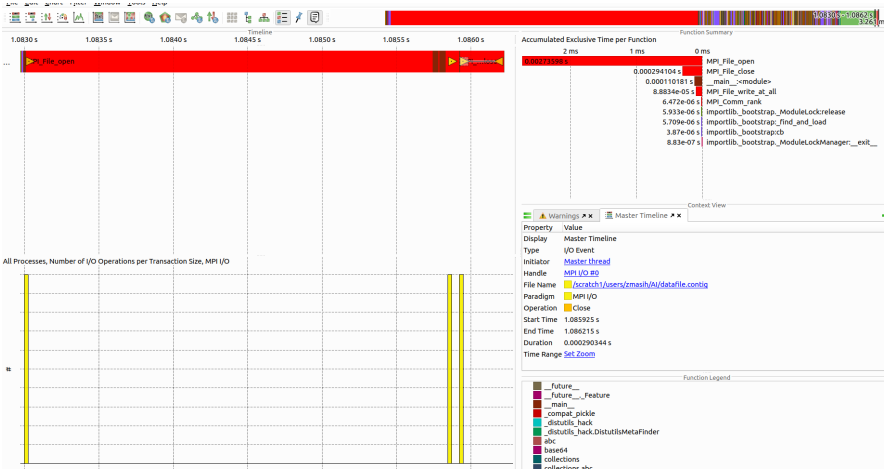
I/O summary


Timeline
1.0830 s
1.0835 s
1.0840 s
1.0845 s
1.0850 s
1.0855 s
1.0860 s
MPI_File_open
MPI_File_close
All Processes, Number of I/O Operations per Transaction Size, MPI I/O
Function Summary
Accumulated Exclusive Time per Function
2 ms
1 ms
0 ms
0.00273598 s MPI_File_open
0.000294104 s MPI_File_close
0.000110181 s __main__:<module>
8.8834e-05 s MPI_File_write_at_all
6.472e-06 s MPI_Comm_rank
5.933e-06 s importlib._bootstrap._ModuleLock:release
5.709e-06 s importlib._bootstrap:_find_and_load
3.87e-06 s importlib._bootstrap:cb
8.83e-07 s importlib._bootstrap._ModuleLockManager:__exit__
Context View
Warnings
Master Timeline
Property Value
Display Master Timeline
Type I/O Event
Initiator Master thread
Handle MPI I/O #0
File Name /scratch1/users/zmasih/AI/datafile.contig
Paradigm MPI I/O
Operation Open
Flag Create (creation)
Flag Write only (access)
Start Time 1.083007 s
End Time 1.085736 s
Duration 0.00272893 s
Function Legend
__future__
__future__._Feature
__main__
_compat_pickle
_distutils_hack
_distutils_hack.DistutilsMetaFinder
abc
base64
collections

I/O summary


Timeline
1.0830 s
1.0835 s
1.0840 s
1.0845 s
1.0850 s
1.0855 s
1.0860 s
MPI_File_open
MPI_File_close
All Processes, Number of I/O Operations per Transaction Size, MPI I/O
Function Summary
Accumulated Exclusive Time per Function
2 ms
1 ms
0 ms
0.00273598 s MPI_File_open
0.000294104 s MPI_File_close
0.000110181 s __main__:<module>
8.8834e-05 s MPI_File_write_at_all
6.472e-06 s MPI_Comm_rank
5.933e-06 s importlib._bootstrap._ModuleLockrelease
5.709e-06 s importlib._bootstrap._find_and_load
3.87e-06 s importlib._bootstrapcb
8.83e-07 s importlib._bootstrap._ModuleLockManager:__exit__
Context View
Warnings
Master Timeline
Property Value
Display Master Timeline
Type I/O Event
Initiator Master thread
Handle MPI I/O #0
File Name /scratch1/users/zmasih/AI/datafile.contig
Paradigm MPI I/O
Operation Write
Flag Collective (operation)
Start Time 1.085848 s
End Time 1.085904 s
Duration 5.6009e-05 s
Time Range Set Zoom
Function Legend
__future__
__future__.Feature
__main__
_compat_pickle
_distutils_hack
_distutils_hack.DistutilsMetaFinder
abc
base64
collections

I/O summary


Timeline
1.0830 s
1.0835 s
1.0840 s
1.0845 s
1.0850 s
1.0855 s
1.0860 s
MPI_File_open
All Processes, Number of I/O Operations per Transaction Size, MPI I/O
Function Summary
Accumulated Exclusive Time per Function
2 ms
1 ms
0 ms
0.00273598 s MPI_File_open
0.000294104 s MPI_File_close
0.000110181 s __main__:<module>
8.8834e-05 s MPI_File_write_at_all
6.472e-06 s MPI_Comm_rank
5.933e-06 s importlib._bootstrap._ModuleLock:release
5.709e-06 s importlib._bootstrap:_find_and_load
3.87e-06 s importlib._bootstrap:cb
8.83e-07 s importlib._bootstrap._ModuleLockManager:__exit__
Context View
Warnings
Master Timeline
Property Value
Display Master Timeline
Type I/O Event
Initiator Master thread
Handle MPI I/O #0
File Name /scratch1/users/zmasih/AI/datafile.contig
Paradigm MPI I/O
Operation Close
Start Time 1.085925 s
End Time 1.086215 s
Duration 0.000290344 s
Time Range Set Zoom
Function Legend
__future__
__future__._Feature
__main__
_compat_pickle
_distutils_hack
_distutils_hack.DistutilsMetaFinder
abc
base64
collections

I/O summary

Context View

Trace Info | Processes | I/O Paradigms | I/O Handles | Collective Operation Types | Attribute Value Strings

Property	Value
File	/home/zoya/Documents/Project/ScoreP_Results/Python/Tracing_io_mpi-io.py/with srun-3processes/scorep-20240426_1317_43386083585181808/traces.otf2
Creator	Score-P 8.3
Version	3.0.0
Timer Resolution	435.799717 ps
Warnings	0
Context Counts	
Attribute Value Strings	13
Attributes	21
Collective Operation Communicators	1
Collective Operation Types	2
Communicators	13
Counter Groups	0
Counters	0
Distinct File Paths	1
Files	1
Function Groups	209
Functions	1056
I/O Handles	11
I/O Paradigms	3
Message Tags	0
Message Types	11
Process Group Paradigms	4
Process Groups	7
Processes	3
Source Code Files	128
Source Code Locations	[illegible]
Event Counts	
Total	179001
States	178980
Counter	0
Messages	0
Collectives	12
File I/O	9
File Substrate	POSIX
Trace Compression	NONE
Trace ID	8f490d36b9ff2e4c

I/O summary

Context View

Trace Info | I/O Paradigms | I/O Handles | Collective Operation Types | Attribute Value Strings

Property	Value
Value	--io=posix
Value	--mpp=mpi
Value	-m
Value	/opt/sw/rev/23.12/linux-scientific7-haswell/gcc-11.4.0/anaconda3-2023.09-0-npsf7i/bin/python3.11
Value	/scratch1
Value	Linux
Value	beegfs
Value	beegfs_nodev
Value	machine
Value	mpi-io.py
Value	r31
Value	scorep
Value	shared memory

I/O summary

Context View

Trace Info | I/O Paradigms | I/O Handles | Collective Operation Types | Attribute Value Strings

Property	Value
Value	CREATE_HANDLE
Value	DESTROY_HANDLE

I/O summary

Context View

Trace Info | I/O Paradigms | I/O Handles | Collective Operation Types | Attribute Value Strings

Property	Value
Handle	MPI I/O #0
Handle	POSIX I/O #0
Handle	POSIX I/O #1
Handle	STDERR_FILENO
Handle	STDIN_FILENO
Handle	STDOUT_FILENO
Handle	all open output streams
Handle	stderr
Handle	stdin
Handle	stdout
Handle	sync - commit buffer cache to disk

I/O summary

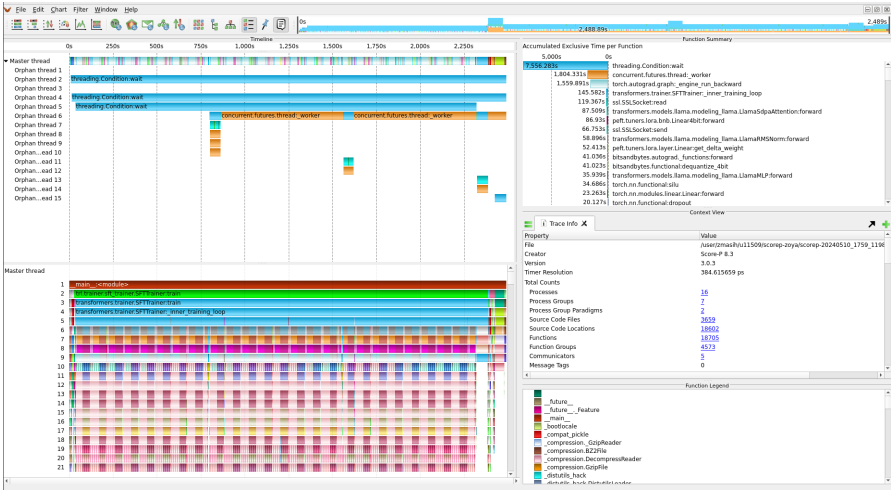
Context View

Trace Info | I/O Paradigms | I/O Handles | Collective Operation Types | Attribute Value Strings

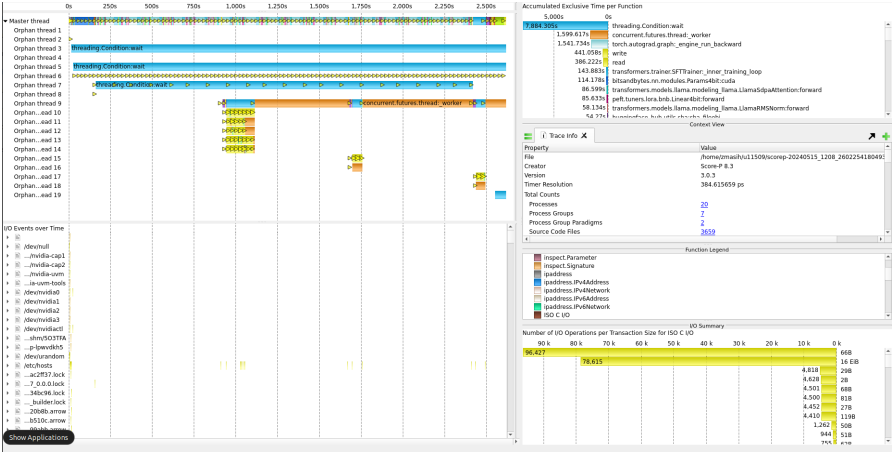
Property	Value
Paradigm	ISO C I/O
Paradigm	MPI I/O
Paradigm	POSIX I/O

Table of Contents

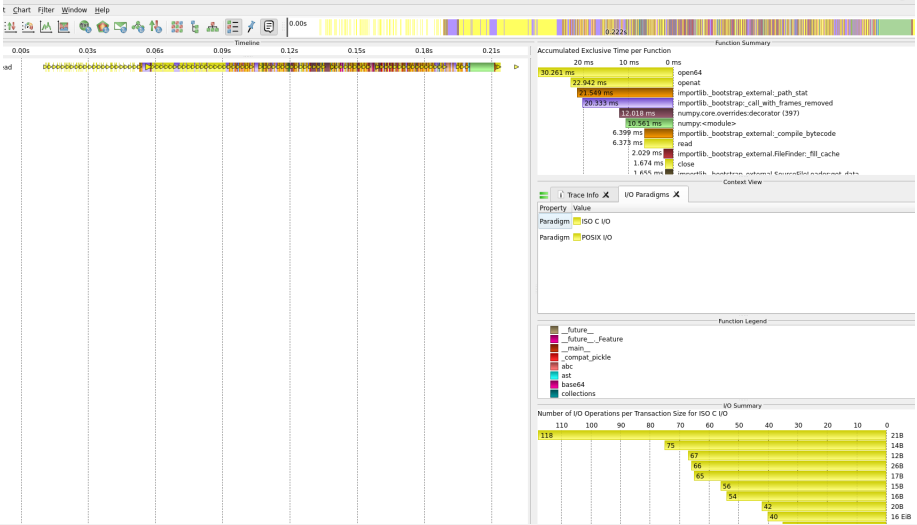
LLM-Fine Tuning



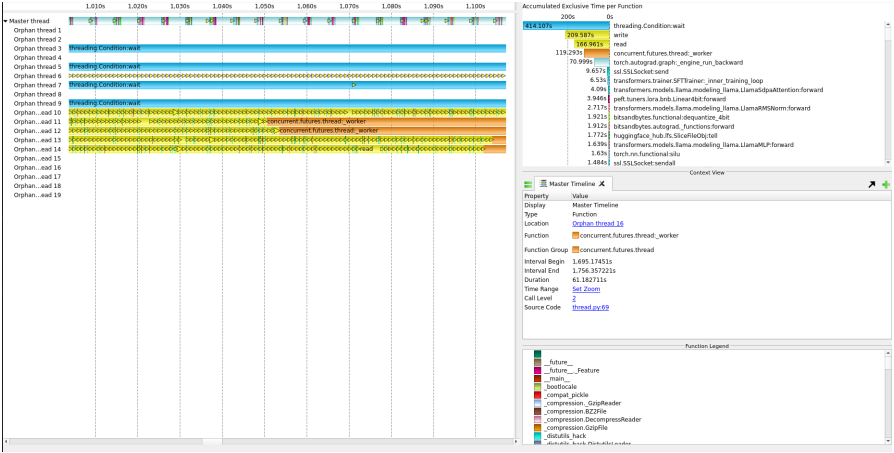
LLM-Fine Tuning



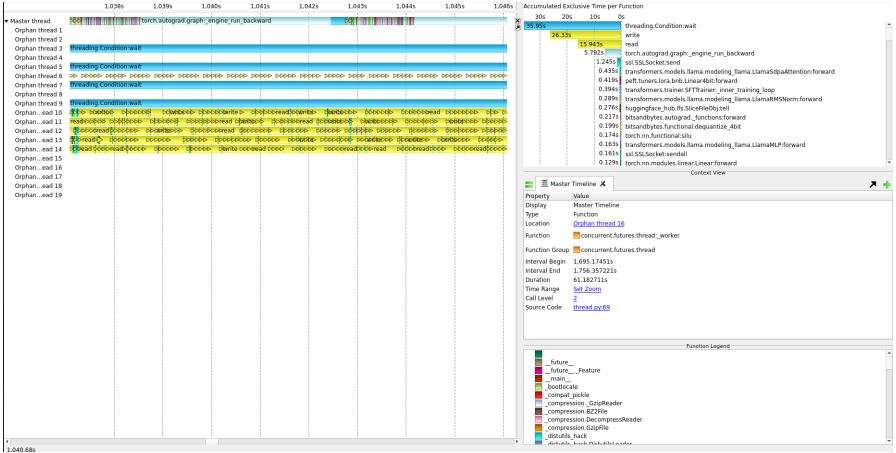
LLM-Fine Tuning



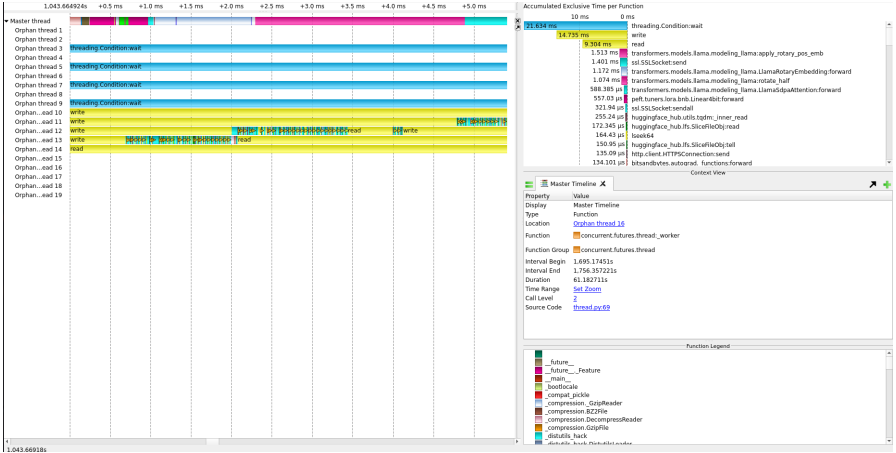
LLM-Fine Tuning



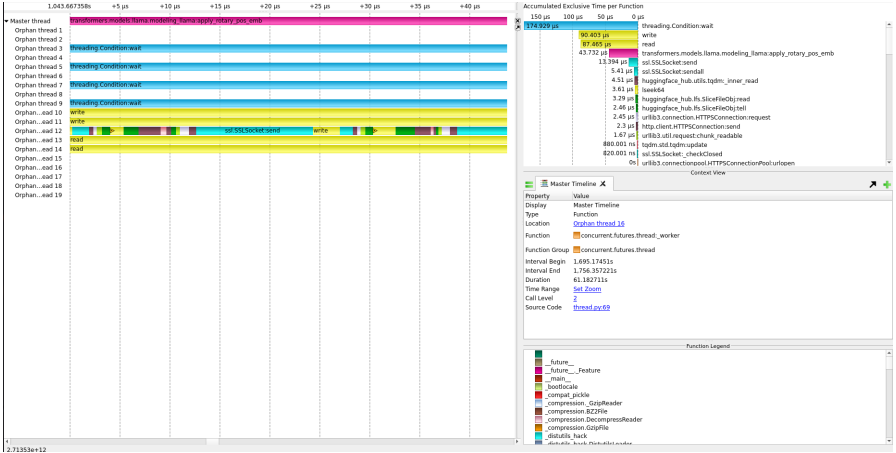
LLM-Fine Tuning



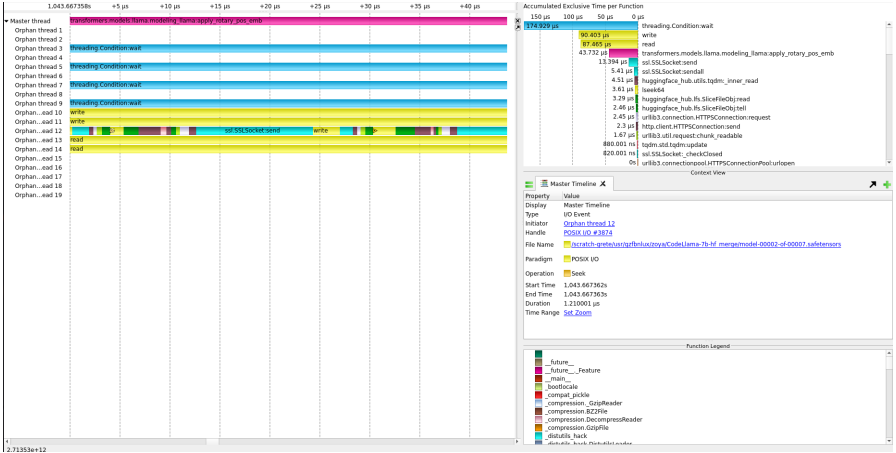
LLM-Fine Tuning



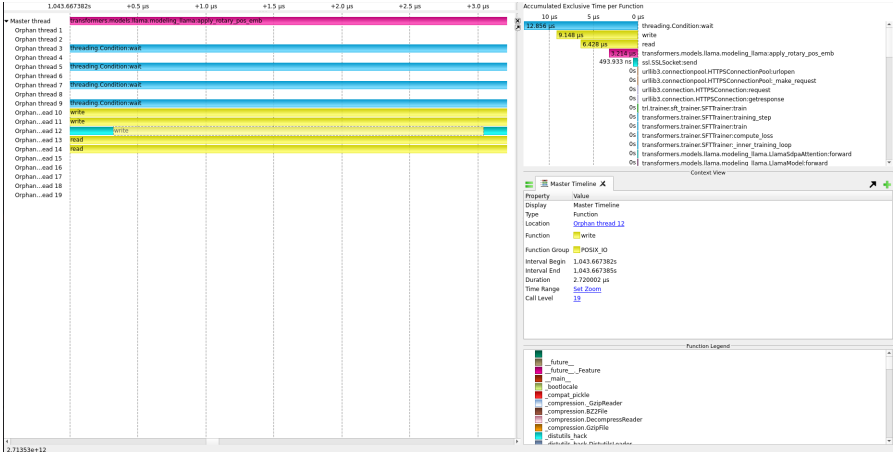
LLM-Fine Tuning



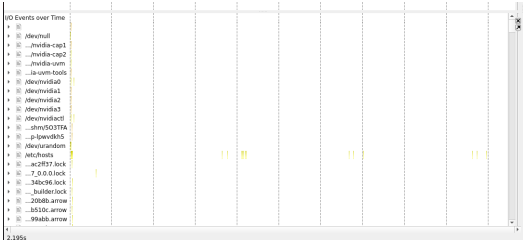
LLM-Fine Tuning



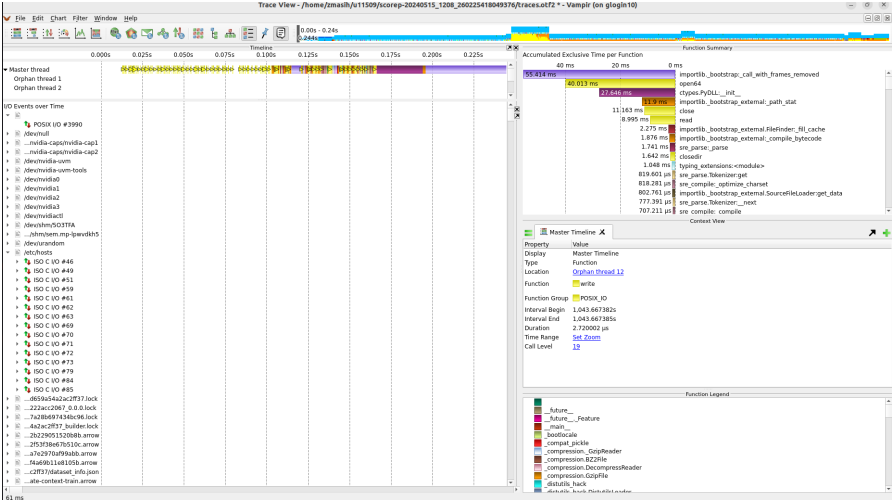
LLM-Fine Tuning



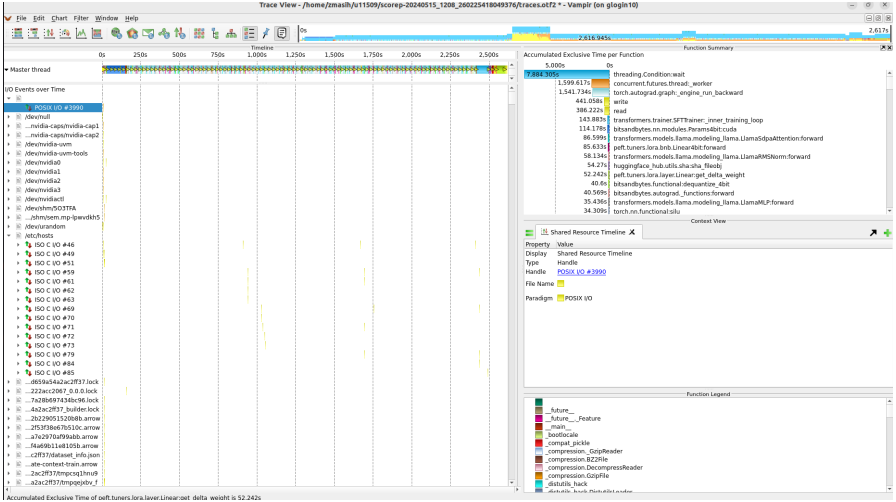
LLM-Fine Tuning



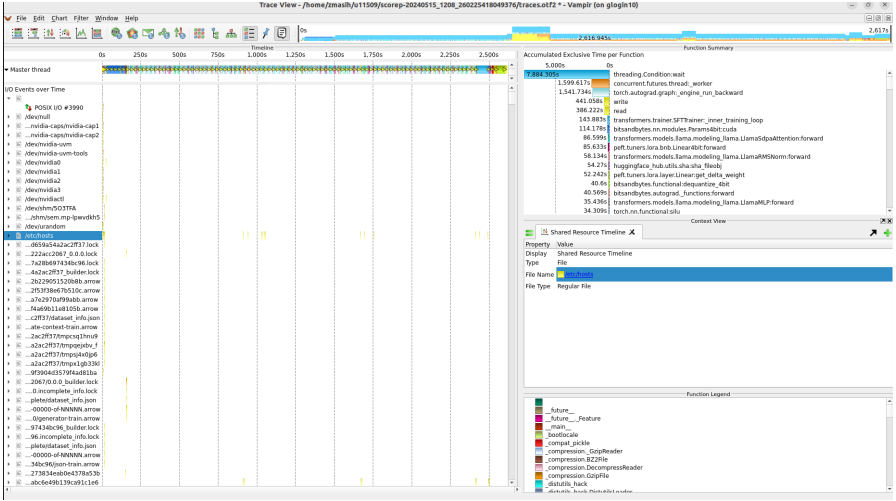
LLM-Fine Tuning



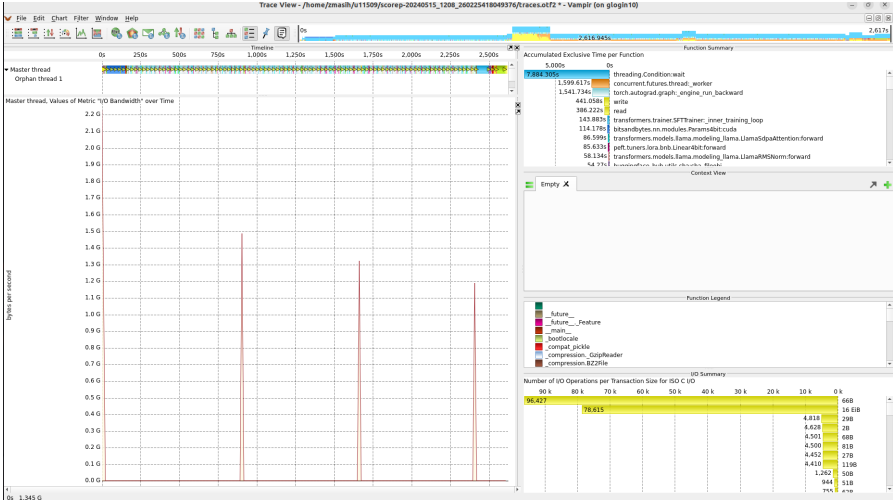
LLM-Fine Tuning



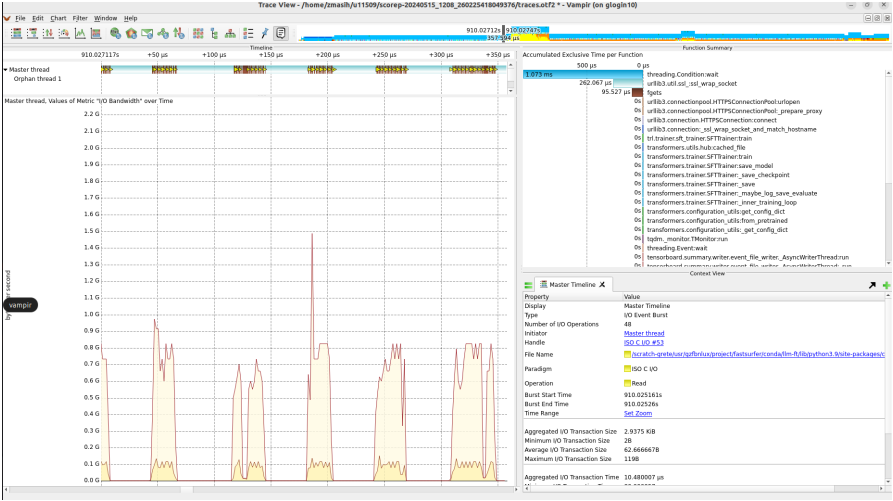
LLM-Fine Tuning



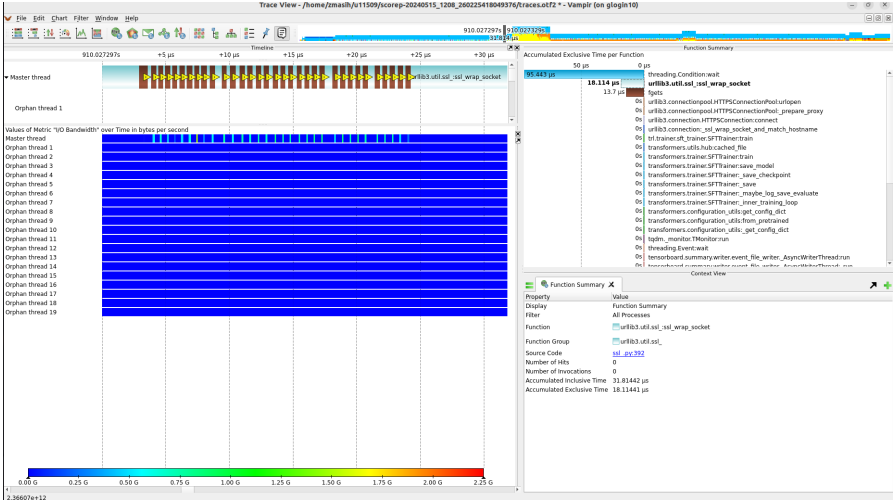
LLM-Fine Tuning



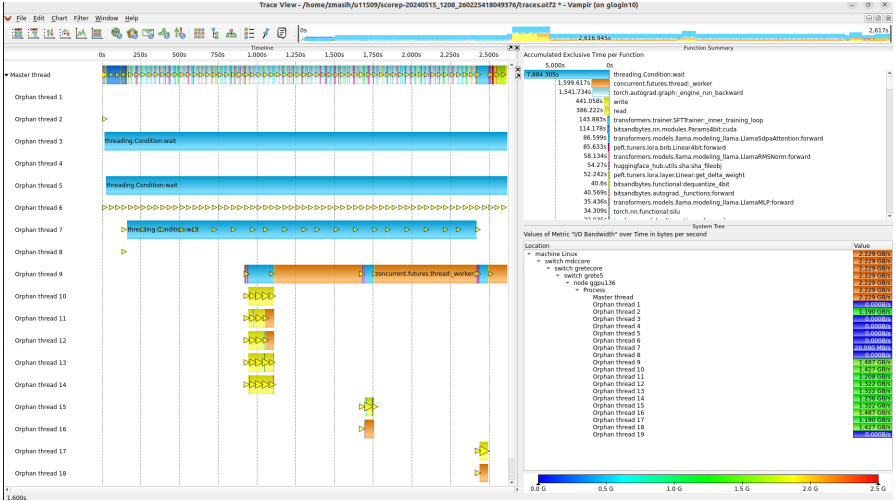
LLM-Fine Tuning



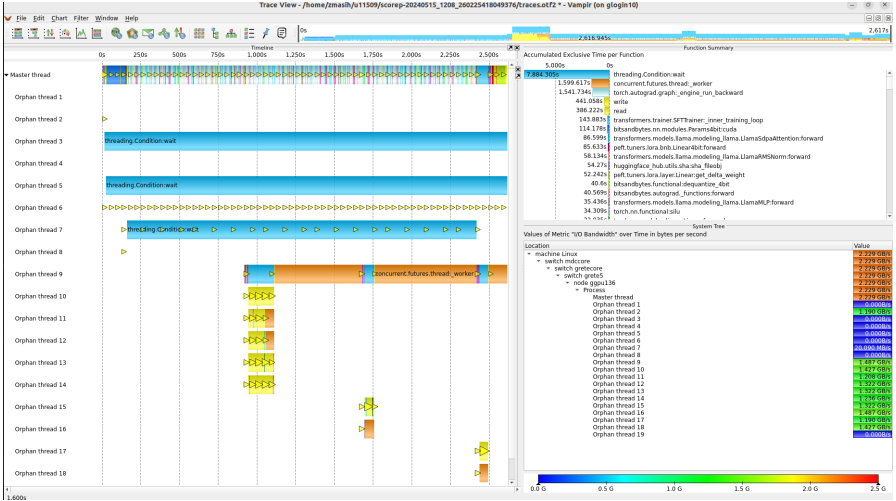
LLM-Fine Tuning



LLM-Fine Tuning



LLM-Fine Tuning



LLM-Fine Tuning

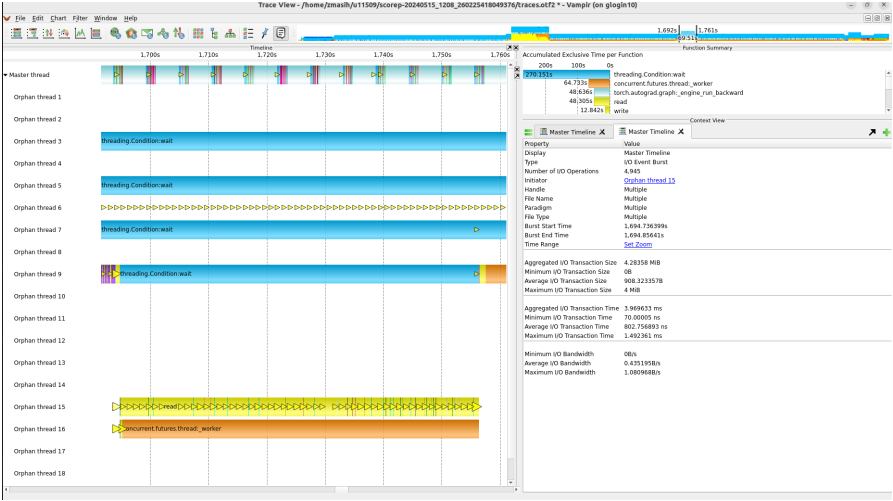


Table of Contents

Challenges

- Generally, X11 forwarding can be slow and inefficient
  - The basic underlying design not well suited to the rising demands of GUI's?
  - RFB (remote frame buffer) protocol used in VNC May help
- ScoreP python-binding currently supports only MPI and SHMEM
- Instrumenting applications is not always straightforward

End

Wish it was helpful

Happy Analysis