# Hatchet: Introduction

Abhinav Bhatele, University of Maryland





# Code-centric performance analysis

- Understanding performance bottlenecks is critical to optimizing (parallel) software
- Profiling and tracing tools help identify parts of code that consume the most time

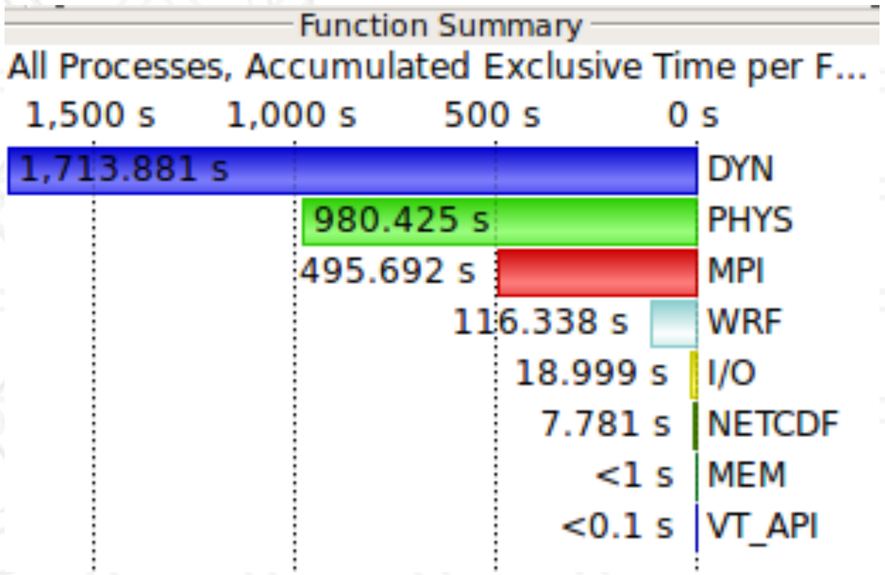
gprof Data

Enter filter text

4 bytes per bucket, each sample counts as 10.000ms

| Name (location) | Samples | Calls | Time/Call | % Time |
|---|---|---|---|---|
| Summary | 2228 | | | 100.0% |
| calc.c | 590 | | | 26.48% |
| copy.c | 0 | | | 0.0% |
| diag.c | 25 | | | 1.12% |
| main.c | 0 | | | 0.0% |
| time.c | 653 | | | 29.31% |
| tstep.c | 958 | | | 43.0% |
| tstep | 958 | 10000 | 957.999us | 43.0% |
| tstep (tstep.c:47) | 1 | | | 0.04% |
| tstep (tstep.c:48) | 62 | | | 2.78% |
| tstep (tstep.c:49) | 46 | | | 2.06% |
| tstep (tstep.c:50) | 46 | | | 2.06% |
| tstep (tstep.c:51) | 48 | | | 2.15% |
| tstep (tstep.c:58) | 101 | | | 4.53% |
| tstep (tstep.c:59) | 135 | | | 6.06% |
| tstep (tstep.c:60) | 120 | | | 5.39% |
| tstep (tstep.c:61) | 126 | | | 5.66% |
| tstep (tstep.c:66) | 3 | | | 0.13% |
| tstep (tstep.c:67) | 108 | | | 4.85% |
| tstep (tstep.c:68) | 63 | | | 2.83% |
| tstep (tstep.c:69) | 43 | | | 1.93% |
| tstep (tstep.c:70) | 56 | | | 2.51% |
| worker.c | 2 | | | 0.09% |

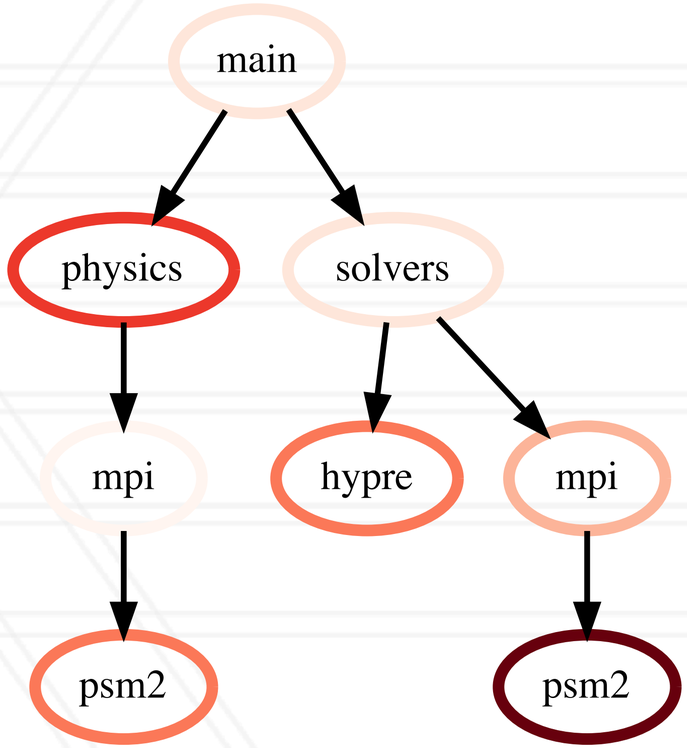
Gprof data in hpctView



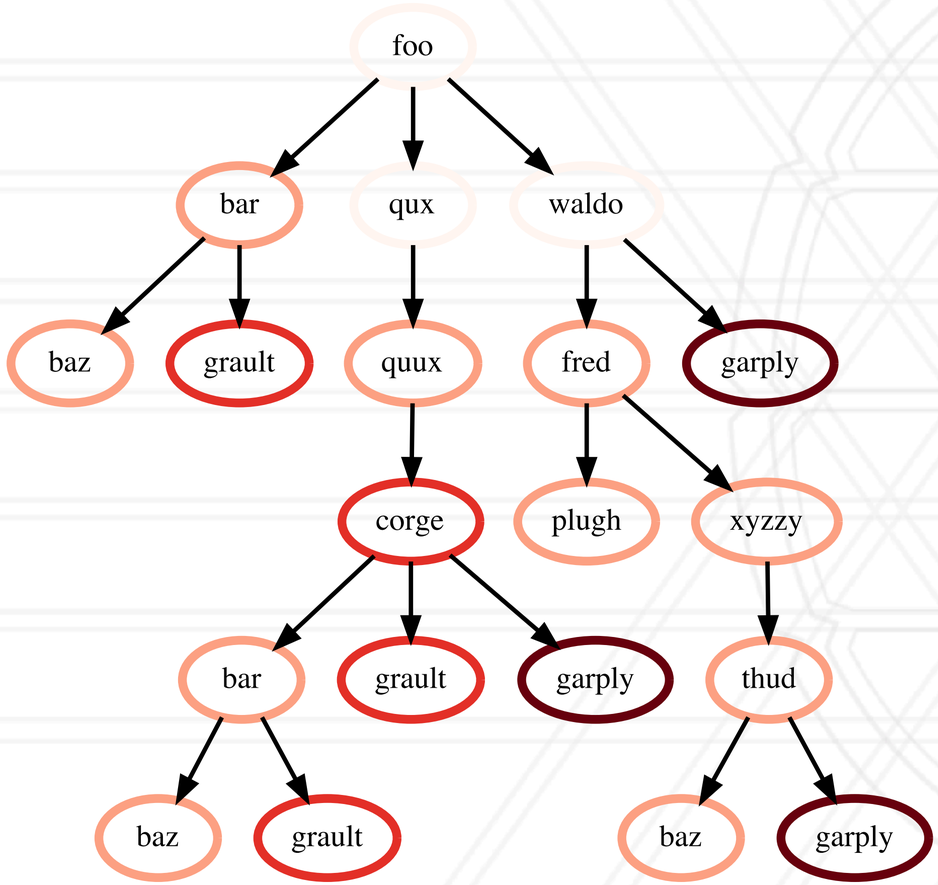

Trace data in Vampir

# Attributing performance (time) to different calling contexts

- Requires reasonable understanding of code structure
- Sophisticated profilers can attribute time to calling contexts: where was a function called from?
- Many measurement tools:
  - HPCToolkit, Caliper, Score-P, TAU, Gprof, Callgrind, perf, Timemory
  - cProfile, pyinstrument

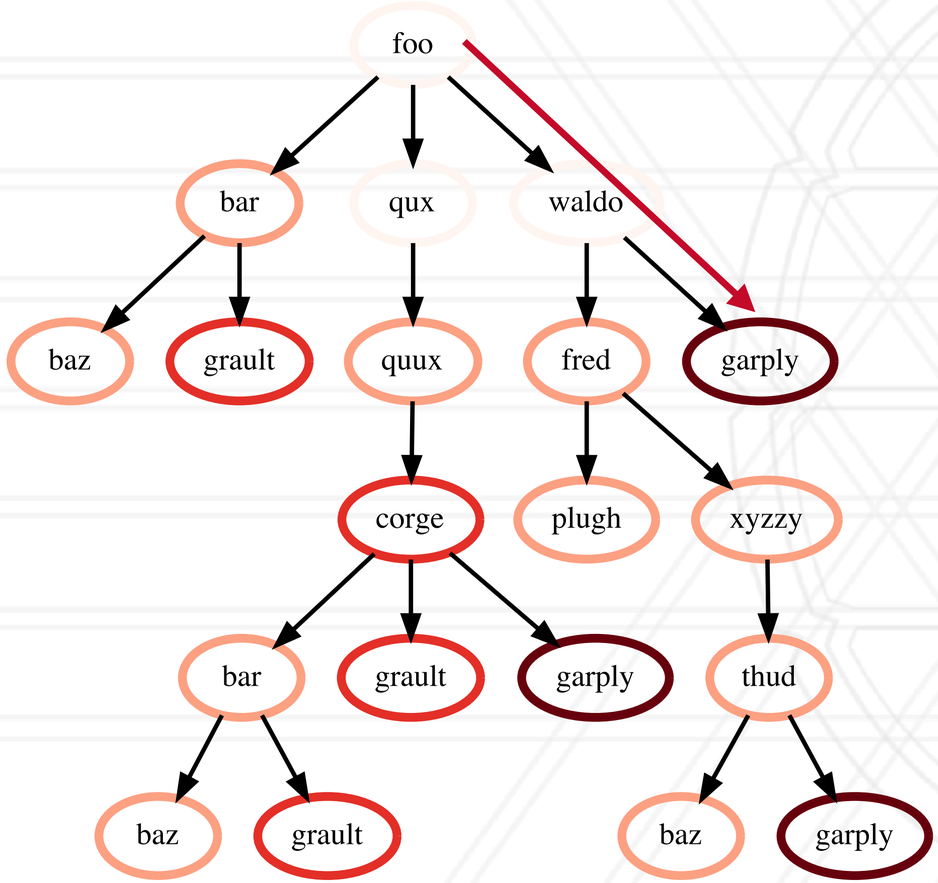


# Calling context trees, call graphs, …



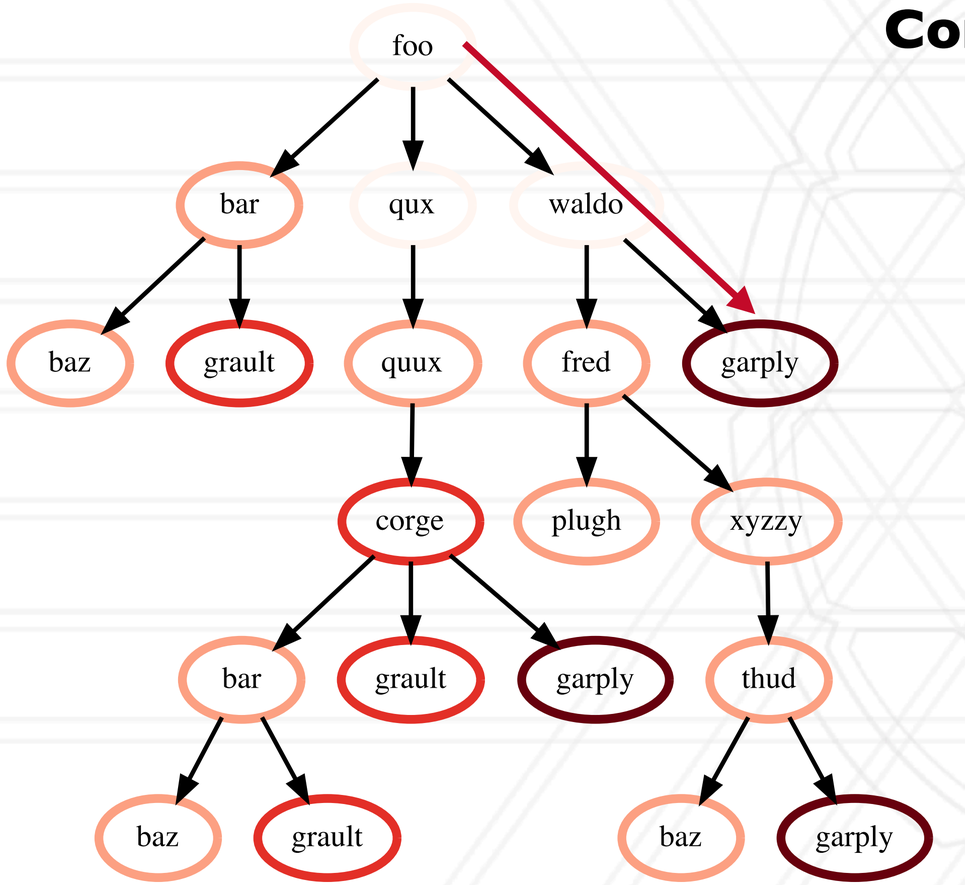

Calling context tree (CCT)

# Calling context trees, call graphs, …




Calling context tree (CCT)

# Calling context trees, call graphs, …

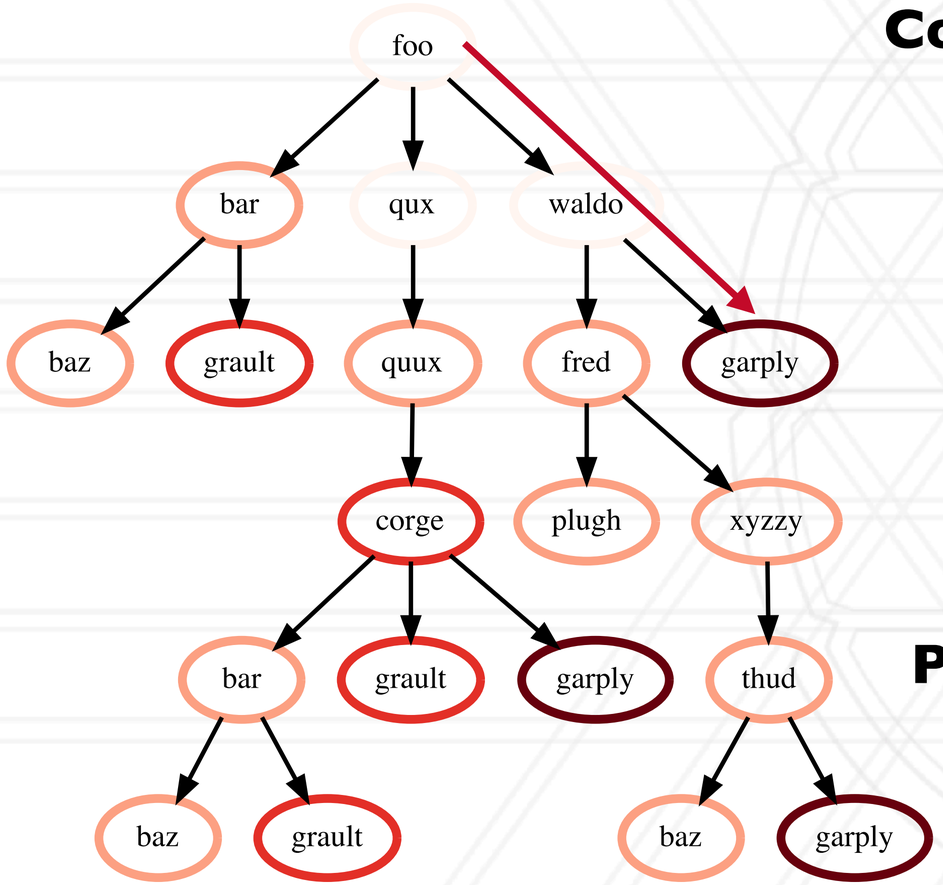



Calling context tree (CCT)

**Contextual information**

```
File
Line number
Function name
Callpath
Load module
Process ID
Thread ID
```

# Calling context trees, call graphs, …




Calling context tree (CCT)

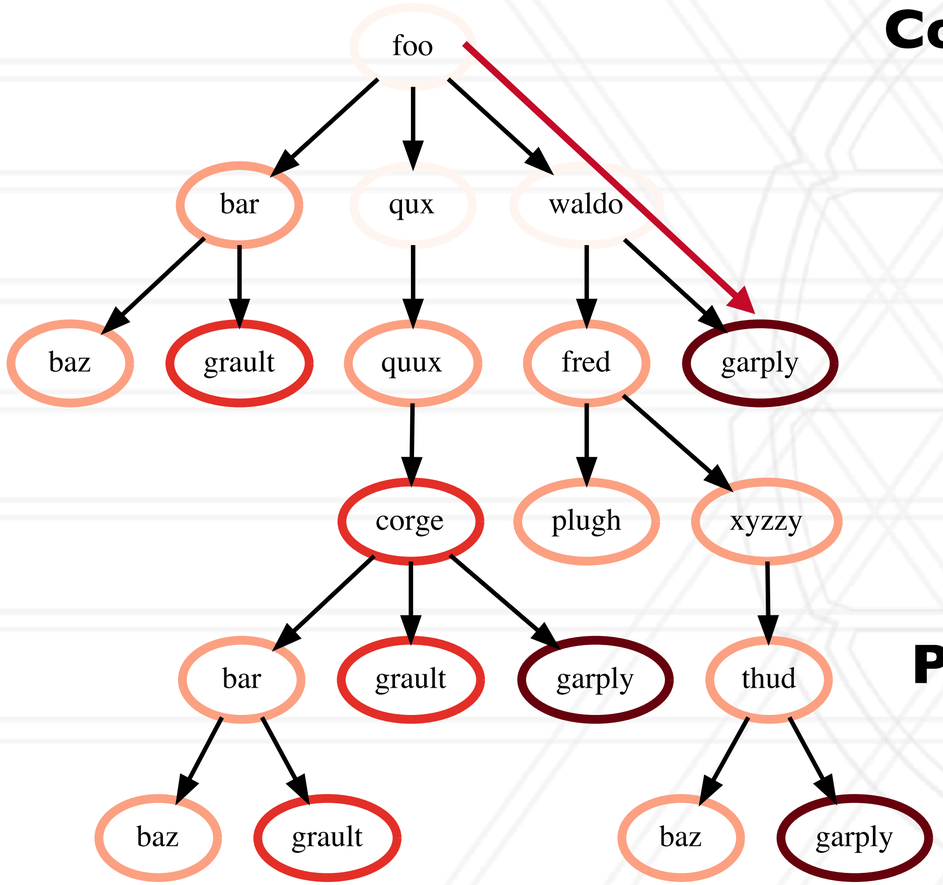
## Contextual information

- File
- Line number
- Function name
- Callpath
- Load module
- Process ID
- Thread ID

## Performance Metrics

- Time
- Flops
- Cache misses

# Calling context trees, call graphs, …



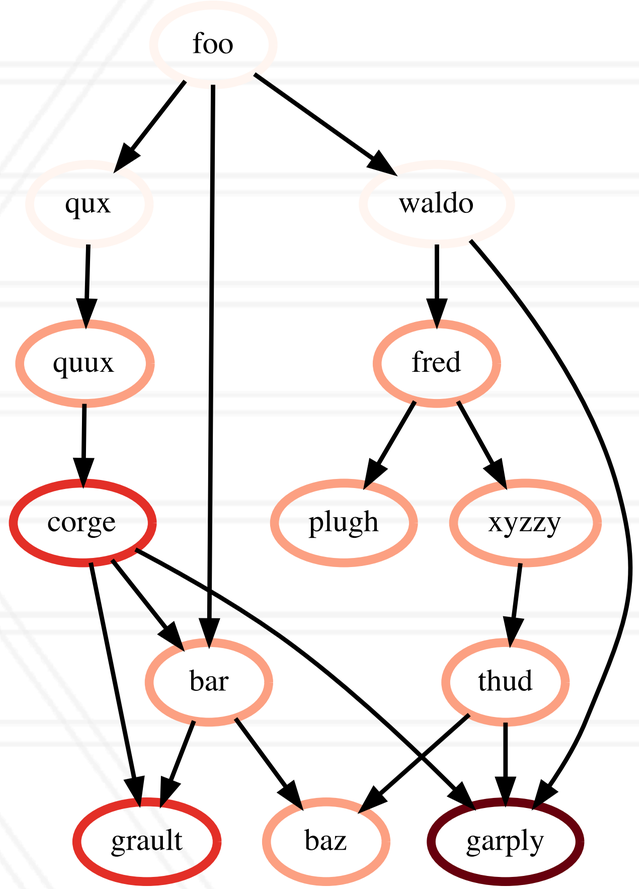

Calling context tree (CCT)

**Contextual information**

- File
- Line number
- Function name
- Callpath
- Load module
- Process ID
- Thread ID

**Performance Metrics**

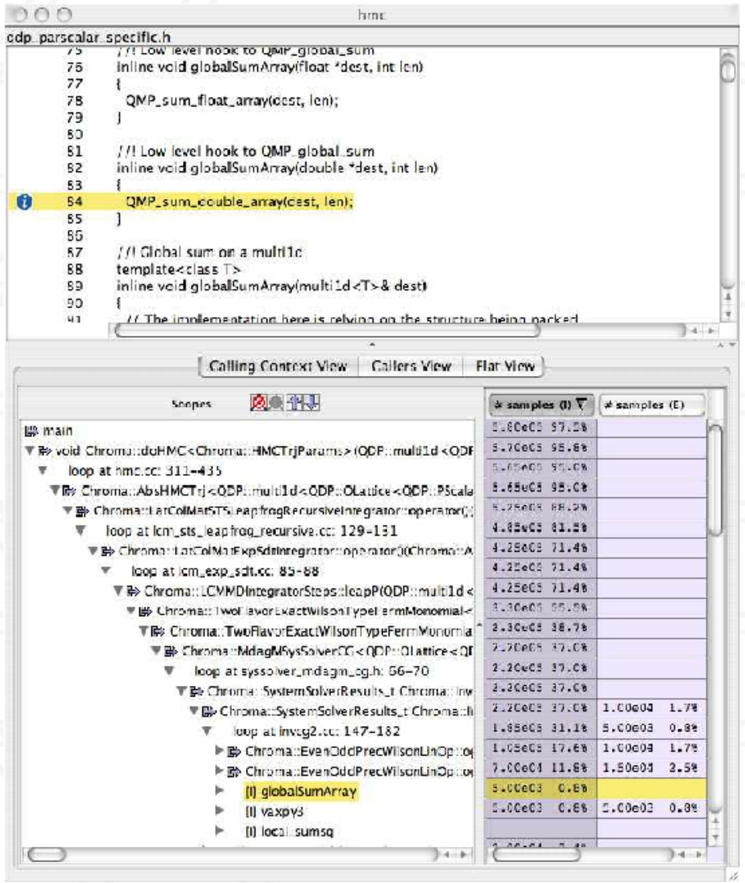
- Time
- Flops
- Cache misses




Call graph

# Limitations of current *analysis* tools

- Support their own unique format(s)
- Limited support for saving or automating analysis
- Most tools only support viewing one dataset at a time
- Lack capabilities to sub-select and focus on specific parts

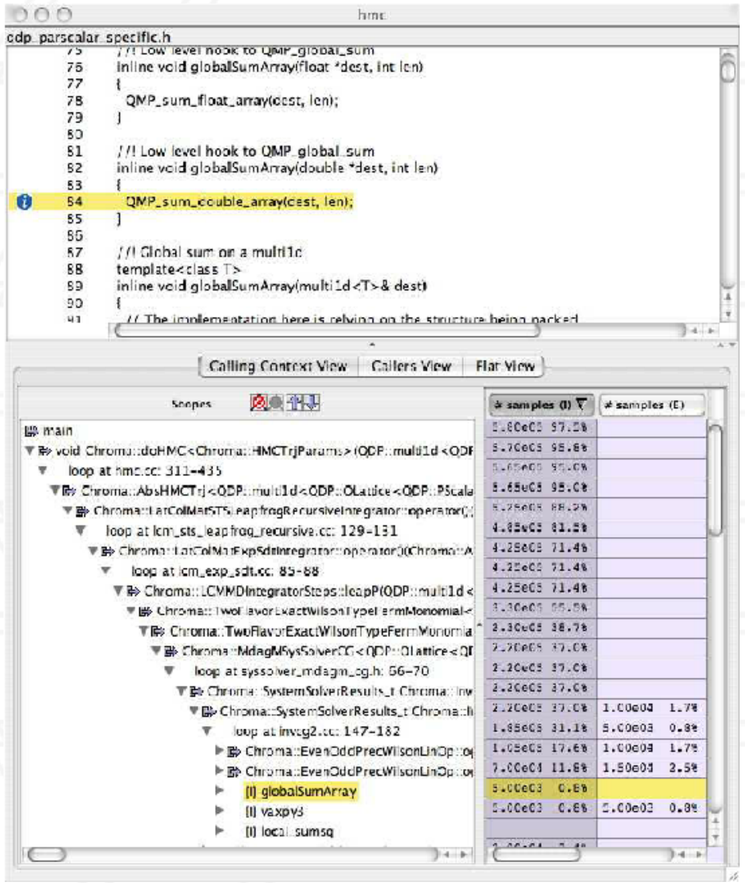



hpcviewer's GUI

# Limitations of current *analysis* tools

- Support their own unique format(s)
- Limited support for saving or automating analysis
- Most tools only support viewing one dataset at a time
- Lack capabilities to sub-select and focus on specific parts

Do not enable programmatic analysis of the data by the end user




hpcviewer's GUI

# The main idea behind Hatchet

- A Python-based library to enable programmatic analysis
- Creates an in-memory representation of the graph
- Leverage pandas which supports multi-dimensional tabular datasets
  - Use graph as structured index to index pandas dataframes
- A set of operators to sub-select and/or aggregate profile data
- A set of operators to compare multiple datasets

# Pandas and dataframes

# Pandas and dataframes

- Pandas is an open-source Python library for data analysis

# Pandas and dataframes

- Pandas is an open-source Python library for data analysis
- Dataframe: two-dimensional tabular data structure
  - Supports many operations borrowed from SQL databases

Columns

Rows

| | node | name | time (inc) | time |
|---|---|---|---|---|
| **0** | {'name': 'main'} | main | 200.0 | 10.0 |
| **1** | {'name': 'physics'} | physics | 60.0 | 40.0 |
| **2** | {'name': 'mpi'} | mpi | 20.0 | 5.0 |
| **3** | {'name': 'psm2'} | psm2 | 15.0 | 30.0 |
| **4** | {'name': 'solvers'} | solvers | 100.0 | 10.0 |
| **5** | {'name': 'hypre'} | hypre | 65.0 | 30.0 |
| **6** | {'name': 'mpi'} | mpi | 35.0 | 20.0 |
| **7** | {'name': 'psm2'} | psm2 | 25.0 | 60.0 |

# Pandas and dataframes

- Pandas is an open-source Python library for data analysis
- Dataframe: two-dimensional tabular data structure
  - Supports many operations borrowed from SQL databases

Index

Columns

Rows

| | node | name | time (inc) | time |
|---|---|---|---|---|
| **0** | {'name': 'main'} | main | 200.0 | 10.0 |
| **1** | {'name': 'physics'} | physics | 60.0 | 40.0 |
| **2** | {'name': 'mpi'} | mpi | 20.0 | 5.0 |
| **3** | {'name': 'psm2'} | psm2 | 15.0 | 30.0 |
| **4** | {'name': 'solvers'} | solvers | 100.0 | 10.0 |
| **5** | {'name': 'hypre'} | hypre | 65.0 | 30.0 |
| **6** | {'name': 'mpi'} | mpi | 35.0 | 20.0 |
| **7** | {'name': 'psm2'} | psm2 | 25.0 | 60.0 |

# Pandas and dataframes

- Pandas is an open-source Python library for data analysis
- Dataframe: two-dimensional tabular data structure
  - Supports many operations borrowed from SQL databases
- MultiIndex enables working with high-dimensional data in a 2D data structure

Index

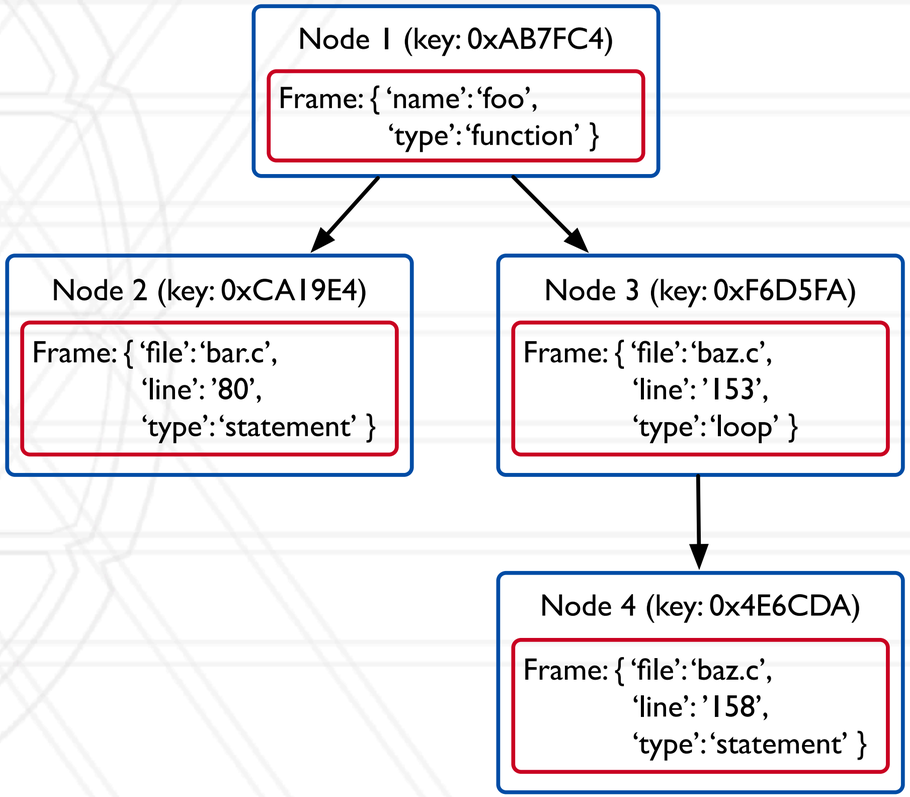
Columns

Rows

| | node | name | time (inc) | time |
|---|---|---|---|---|
| 0 | {'name': 'main'} | main | 200.0 | 10.0 |
| 1 | {'name': 'physics'} | physics | 60.0 | 40.0 |
| 2 | {'name': 'mpi'} | mpi | 20.0 | 5.0 |
| 3 | {'name': 'psm2'} | psm2 | 15.0 | 30.0 |
| 4 | {'name': 'solvers'} | solvers | 100.0 | 10.0 |
| 5 | {'name': 'hypre'} | hypre | 65.0 | 30.0 |
| 6 | {'name': 'mpi'} | mpi | 35.0 | 20.0 |
| 7 | {'name': 'psm2'} | psm2 | 25.0 | 60.0 |

# Canonical data model: *a structured index*

- Structured index is basically an in-memory graph
- Each node is assigned a unique key, which enables using the nodes as the index in the dataframe
- Each node has a *Frame* that describes the code it represents
  - a set of key/value pairs
- Frames don't have a rigid schema
- *Nodes* define the structure and connectivity

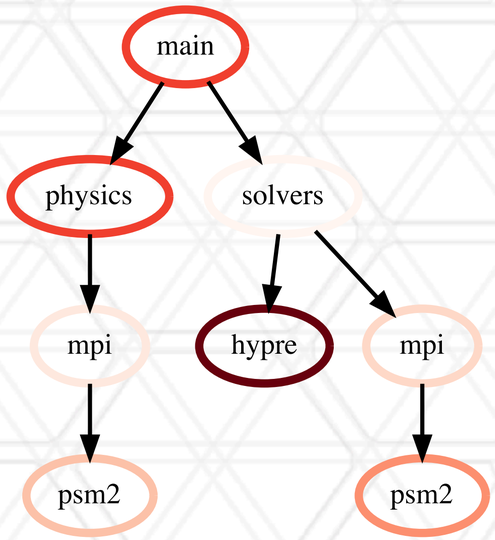


# Central data structure: a *GraphFrame*

- Consists of a structured index graph object and a pandas dataframe
- Graph stores caller-callee relationships
- Dataframe stores all numerical and categorical data

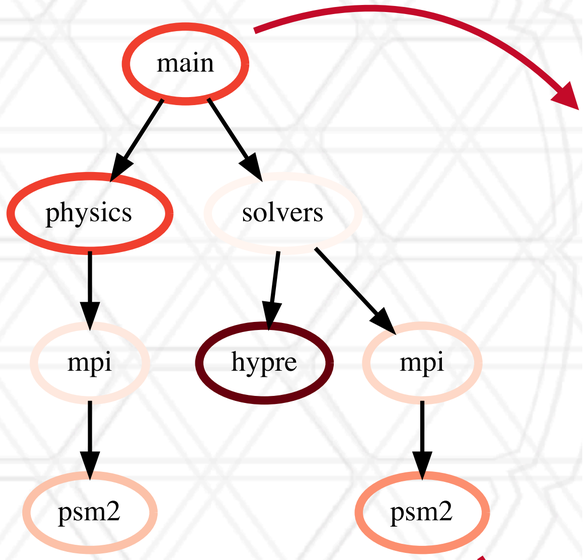
# Central data structure: a *GraphFrame*

- Consists of a structured index graph object and a pandas dataframe
- Graph stores caller-callee relationships
- Dataframe stores all numerical and categorical data



# Central data structure: a *GraphFrame*

- Consists of a structured index graph object and a pandas dataframe
- Graph stores caller-callee relationships
- Dataframe stores all numerical and categorical data




| node | name | nid | node | time | time (inc) |
|---|---|---|---|---|---|
| **main** | main | 0 | main | 40.0 | 200.0 |
| **physics** | physics | 1 | physics | 40.0 | 60.0 |
| **mpi** | mpi | 2 | mpi | 5.0 | 20.0 |
| **psm2** | psm2 | 3 | psm2 | 15.0 | 15.0 |
| **solvers** | solvers | 4 | solvers | 0.0 | 100.0 |
| **hypre** | hypre | 5 | hypre | 65.0 | 65.0 |
| **mpi** | mpi | 6 | mpi | 10.0 | 35.0 |
| **psm2** | psm2 | 7 | psm2 | 25.0 | 25.0 |

# Use of MultiIndex

- When metrics are per MPI process, thread etc….

| node | rank | time (inc) | time | nid | rank | file | line | module |
|---|---|---|---|---|---|---|---|---|
| {'type': 'function', 'name': '<program root>'} | 0 | 999238.0 | 0.0 | 2 | 0 | <unknown file> | 0 | /collab/usr/global/tools/hpctoolkit/chaos_5_x8... |
| | 1 | 999390.0 | 0.0 | 2 | 1 | <unknown file> | 0 | /collab/usr/global/tools/hpctoolkit/chaos_5_x8... |
| {'type': 'function', 'name': '27:MPI_Init'} | 0 | 0.0 | 0.0 | 6 | 0 | <unknown file> | 0 | /collab/usr/global/tools/hpctoolkit/chaos_5_x8... |
| | 1 | 0.0 | 0.0 | 6 | 1 | <unknown file> | 0 | /collab/usr/global/tools/hpctoolkit/chaos_5_x8... |
| {'type': 'function', 'name': 'main'} | 0 | 999238.0 | 0.0 | 4 | 0 | ./src/cpi.c | 19 | cpi |
| | 1 | 999390.0 | 0.0 | 4 | 1 | ./src/cpi.c | 19 | cpi |

# Use of MultiIndex

- When metrics are per MPI process, thread etc….

Index

| node | rank | time (inc) | time | nid | rank | file | line | module |
|---|---|---|---|---|---|---|---|---|
| {'type': 'function', 'name': '<program root>'} | 0 | 999238.0 | 0.0 | 2 | 0 | <unknown file> | 0 | /collab/usr/global/tools/hpctoolkit/chaos_5_x8... |
| | 1 | 999390.0 | 0.0 | 2 | 1 | <unknown file> | 0 | /collab/usr/global/tools/hpctoolkit/chaos_5_x8... |
| {'type': 'function', 'name': '27:MPI_Init'} | 0 | 0.0 | 0.0 | 6 | 0 | <unknown file> | 0 | /collab/usr/global/tools/hpctoolkit/chaos_5_x8... |
| | 1 | 0.0 | 0.0 | 6 | 1 | <unknown file> | 0 | /collab/usr/global/tools/hpctoolkit/chaos_5_x8... |
| {'type': 'function', 'name': 'main'} | 0 | 999238.0 | 0.0 | 4 | 0 | ./src/cpi.c | 19 | cpi |
| | 1 | 999390.0 | 0.0 | 4 | 1 | ./src/cpi.c | 19 | cpi |

# *Immutable* semantics for graph nodes

- Having direct references to graph nodes in the dataframe is risky
  - In particular when graph nodes are shared by multiple graphframes
- Any operation that modifies graph nodes in place creates a new GraphFrame and a new graph index
- Implemented using copy-on-write semantics

# Reading in an input dataset

**Installing hatchet**

```
$ pip install hatchet
```

- HPCToolkit, Caliper, gprof
- Pyinstrument, cprofile
- String literal
- In progress: Timemory, TAU, cube

**https://github.com/hatchet/hatchet**

# Reading in an input dataset

**Installing hatchet**

```
$ pip install hatchet
```

- HPCToolkit, Caliper, gprof
- Pyinstrument, cprofile
- String literal
- In progress: Timemory, TAU, cube

**https://github.com/hatchet/hatchet**

```
import hatchet as ht

if __name__ == "__main__":
    dirname = "hpctoolkit-database"
    gf = ht.GraphFrame.from_hpctoolkit(dirname)
```

# Reading in an input dataset

**Installing hatchet**

```
$ pip install hatchet
```

- HPCToolkit, Caliper, gprof
- Pyinstrument, cprofile
- String literal
- In progress: Timemory, TAU, cube

**https://github.com/hatchet/hatchet**

Contribute a reader to hatchet!

```
import hatchet as ht

if __name__ == "__main__":
    dirname = "hpctoolkit-database"
    gf = ht.GraphFrame.from_hpctoolkit(dirname)
```


Abhinav Bhatele

Parallel Software and Systems Group

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu

UNIVERSITY OF MARYLAND