High Performance Computing Systems (CMSC714)

Lecture 6: Measurement Tools

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

Summary of last lecture

- Shared-memory programming and OpenMP
- Fork-join parallelism
- OpenMP vs MPI: ease of programming, performance

Performance analysis

- Parallel performance of a program might not be what the developer expects
- How do we find performance bottlenecks?
- Two parts to performance analysis: measurement and analysis/visualization
- Simplest tool: timers in the code and printf

Using timers

```
double start, end;
double phase1, phase2, phase3;

start = MPI_Wtime();
 ... phase1 code ...
end = MPI_Wtime();
phase1 = end - start;

start = MPI_Wtime();
 ... phase2 ...
end = MPI_Wtime();
phase2 = end - start;

start = MPI_Wtime();
 ... phase3 ...
end = MPI_Wtime();
phase3 = end - start;
```

Using timers

```
double start, end;
double phase1, phase2, phase3;

start = MPI_Wtime();
 ... phase1 code ...
end = MPI_Wtime();
phase1 = end - start;

start = MPI_Wtime();
 ... phase2 ...
end = MPI_Wtime();
phase2 = end - start;

start = MPI_Wtime();
 ... phase3 ...
end = MPI_Wtime();
phase3 = end - start;
```

Phase 1 took 2.45 s

Phase 2 took 11.79 s

Phase 3 took 4.37 s

Performance Tools

- Tracing tools
  - Capture entire execution trace
  - Vampir, Score-P
- Profiling tools
  - Provide aggregated information
  - Typically use statistical sampling
  - Gprof, pyinstrument, cprofile
- Many tools can do both
  - TAU, HPCToolkit, Projections

Metrics recorded

- Counts of function invocations
- Time spent in code
- Number of bytes sent
- Hardware counters
- To fix performance problems — we need to connect metrics to source code

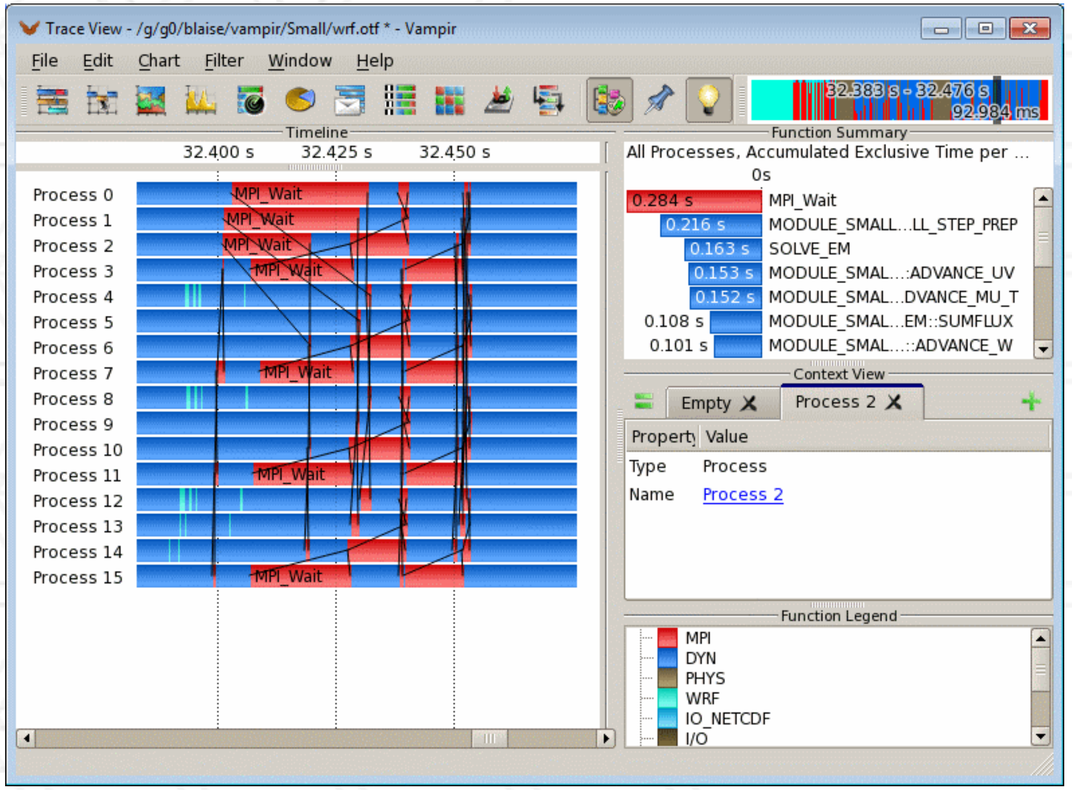
Tracing tools

- Record all the events in the program with timestamps
- Events: function calls, MPI events, etc.

Vampir visualization: https://hpc.llnl.gov/software/development-environment-software/vampir-vampir-server

Tracing tools

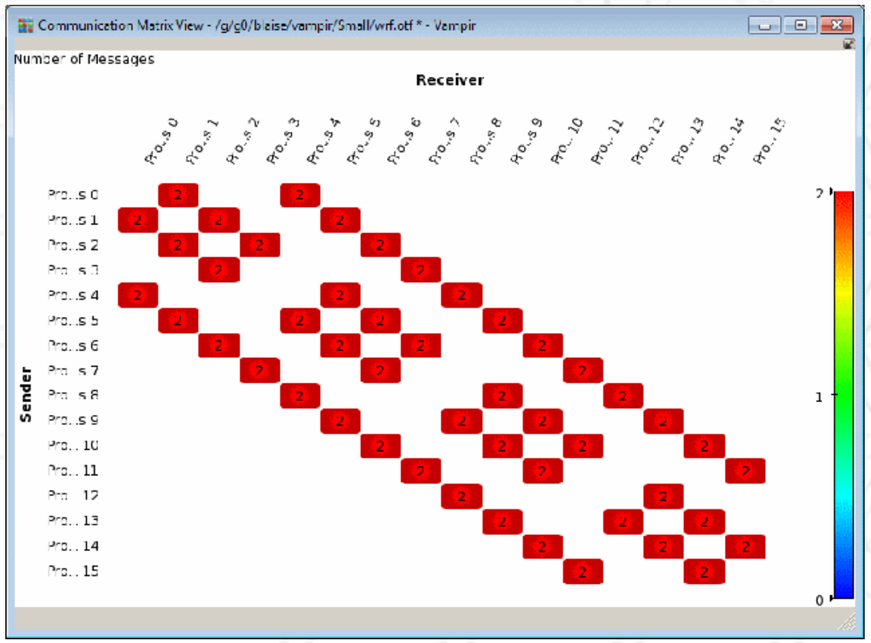
- Record all the events in the program with timestamps
- Events: function calls, MPI events, etc.

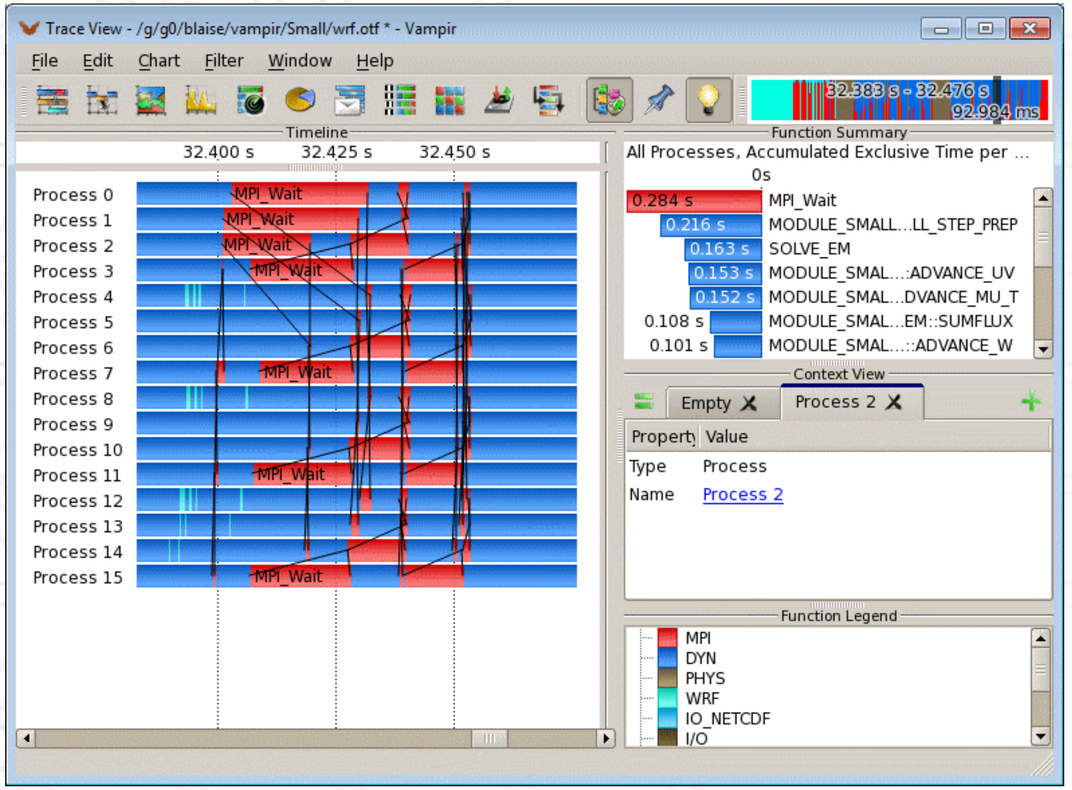


Vampir visualization: https://hpc.llnl.gov/software/development-environment-software/vampir-vampir-server

Tracing tools

- Record all the events in the program with timestamps
- Events: function calls, MPI events, etc.





Vampir visualization: https://hpc.llnl.gov/software/development-environment-software/vampir-vampir-server

Profiling tools

- Ignore the specific times at which events occurred
- Provide aggregate information about different parts of the code
- Examples:
  - Gprof, perf
  - mpiP
  - HPCToolkit, caliper
- Python tools: cprofile, pyinstrument, scalene

gprof Data

Enter filter text

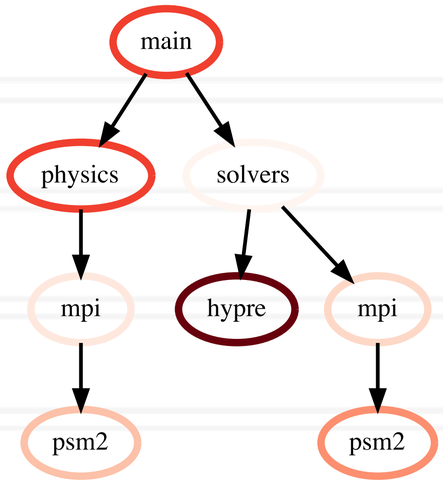
4 bytes per bucket, each sample counts as 10.000ms

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%

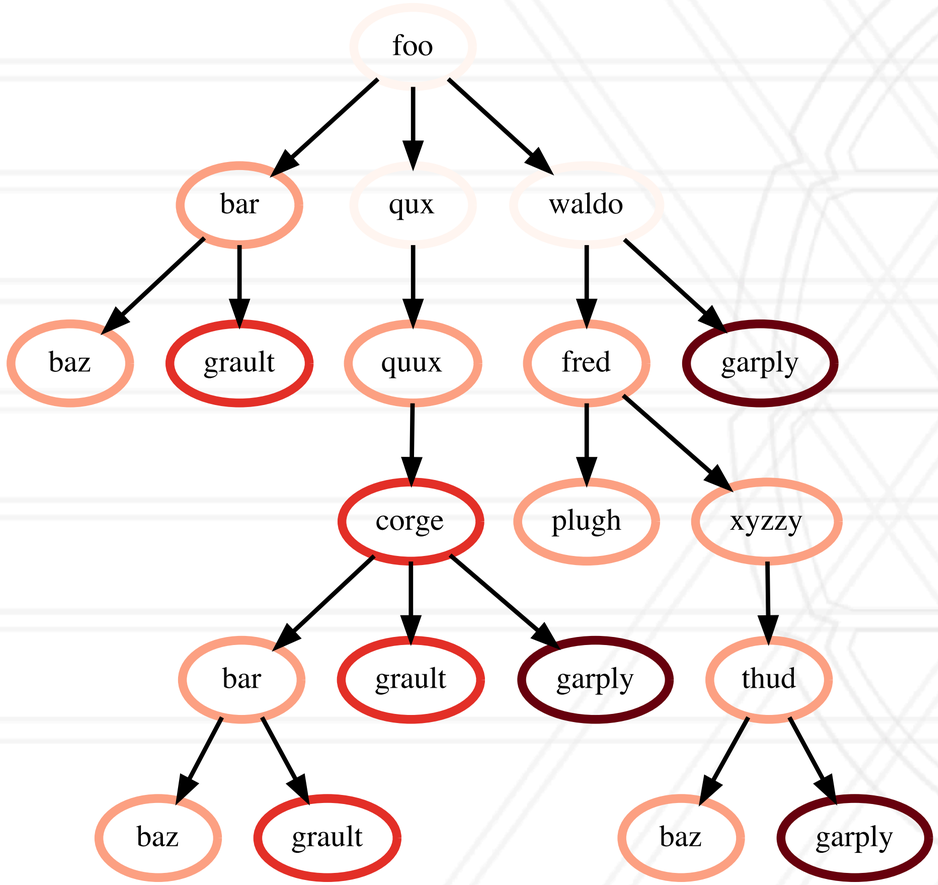
Gprof data in hpctView

Calling contexts, trees, and graphs

- Calling context or call path: Sequence of function invocations leading to the current sample
- Calling context tree (CCT): dynamic prefix tree of all call paths in an execution
- Call graph: merge nodes in a CCT with the same name into a single node but keep caller-callee relationships as arcs

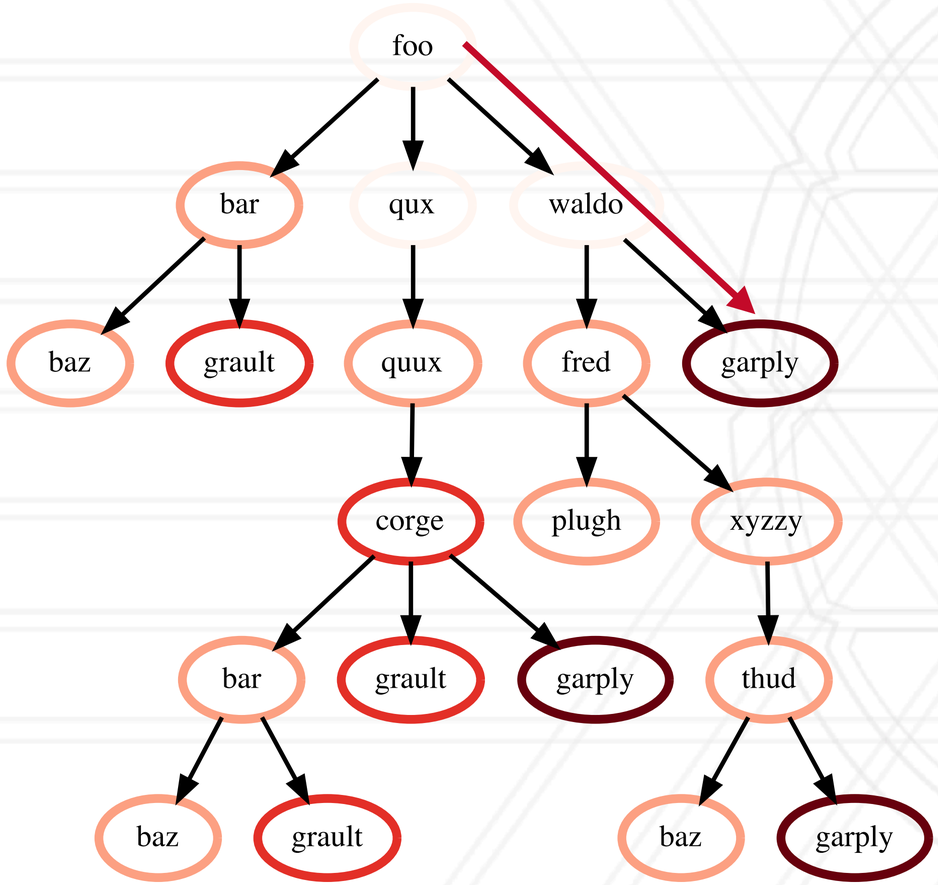


Calling context trees, call graphs, …



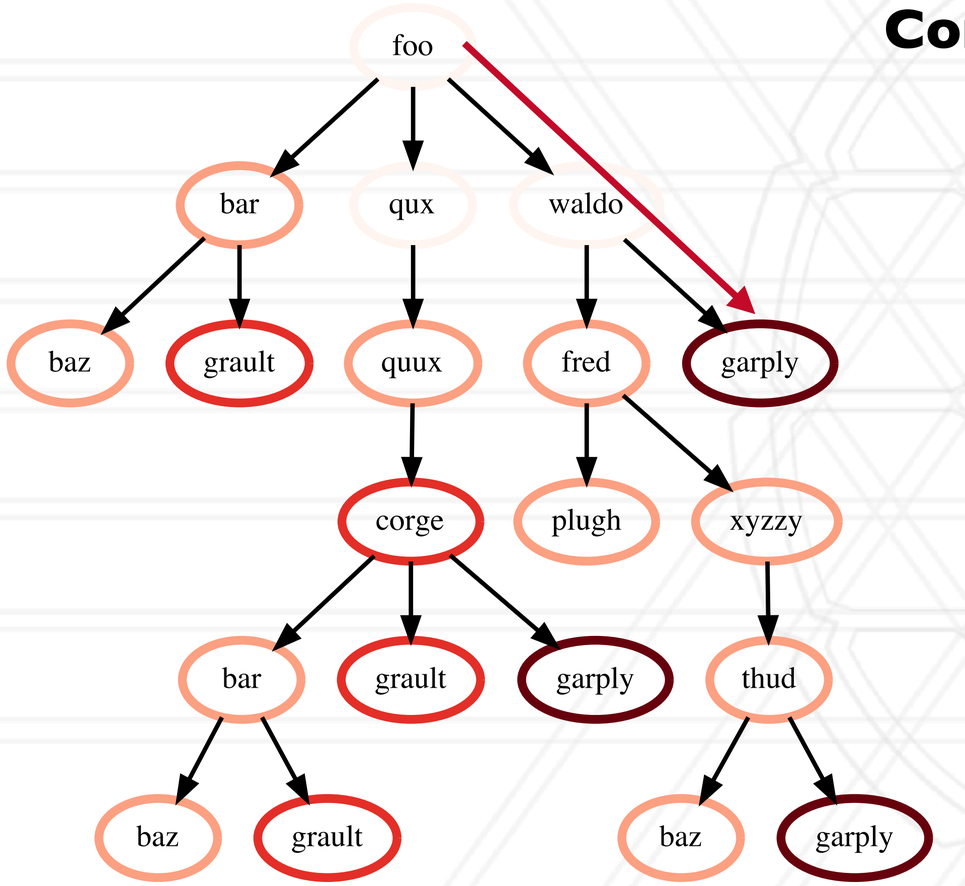
Calling context tree (CCT)

Calling context trees, call graphs, …



Calling context tree (CCT)

Calling context trees, call graphs, …

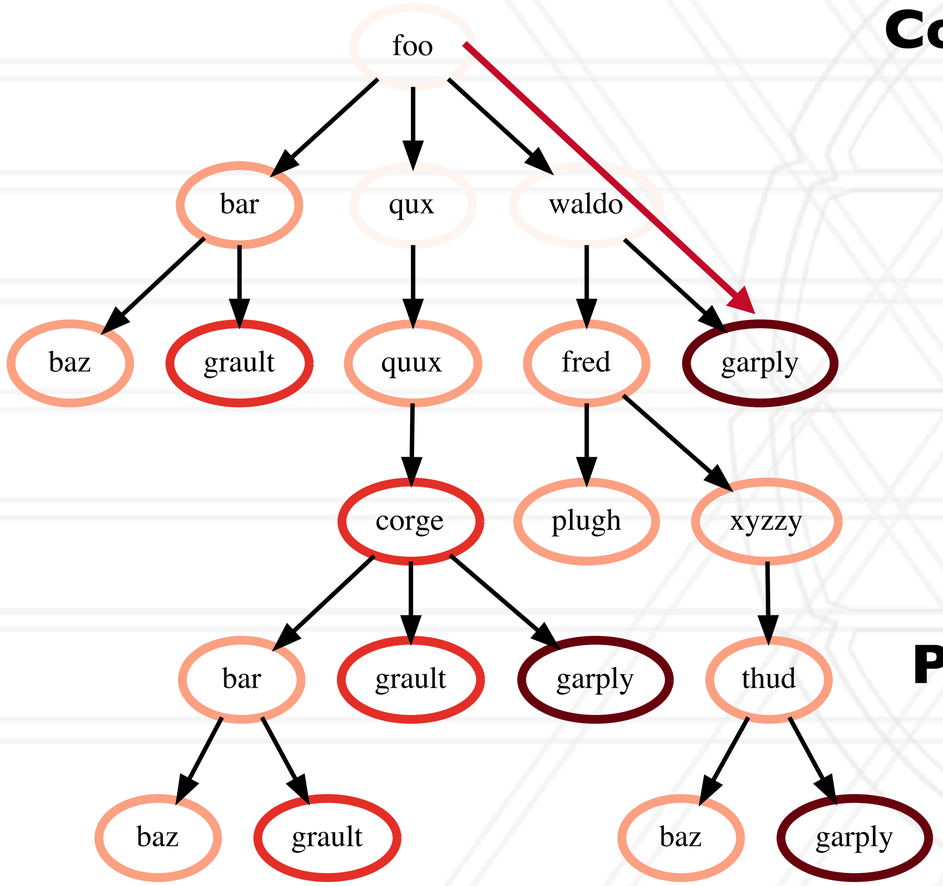


Calling context tree (CCT)

Contextual information

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

Calling context trees, call graphs, …



Calling context tree (CCT)

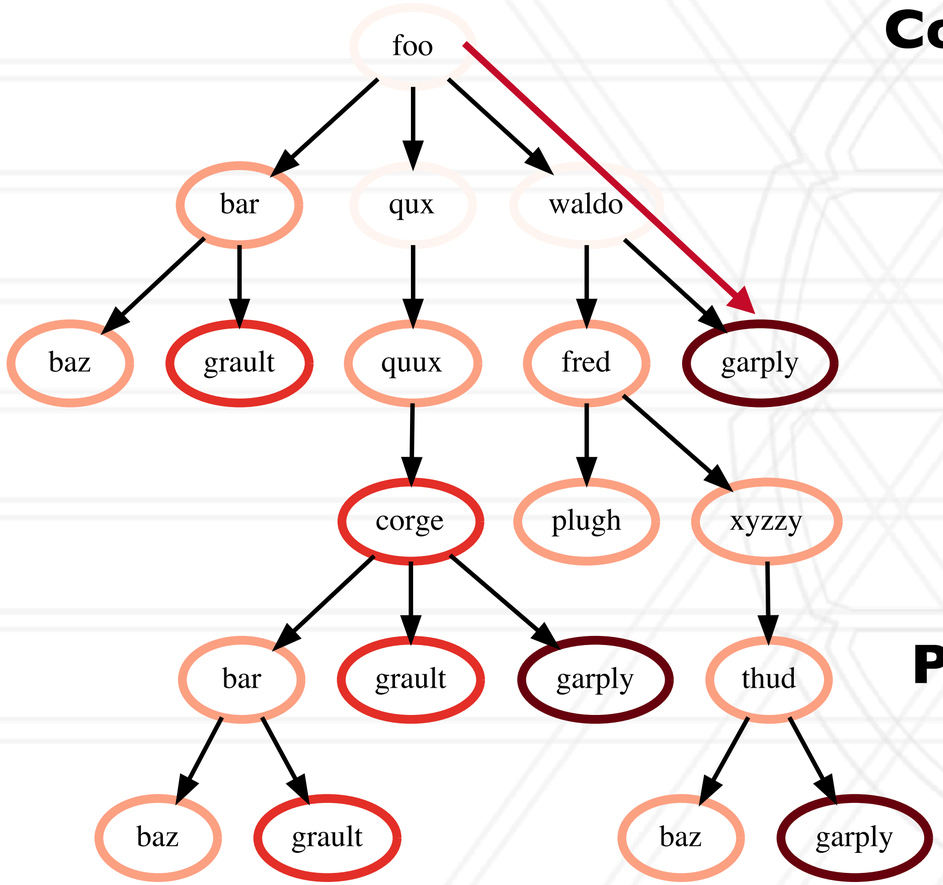
Contextual information

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

Performance Metrics

- Time
- Flops
- Cache misses

Calling context trees, call graphs, …



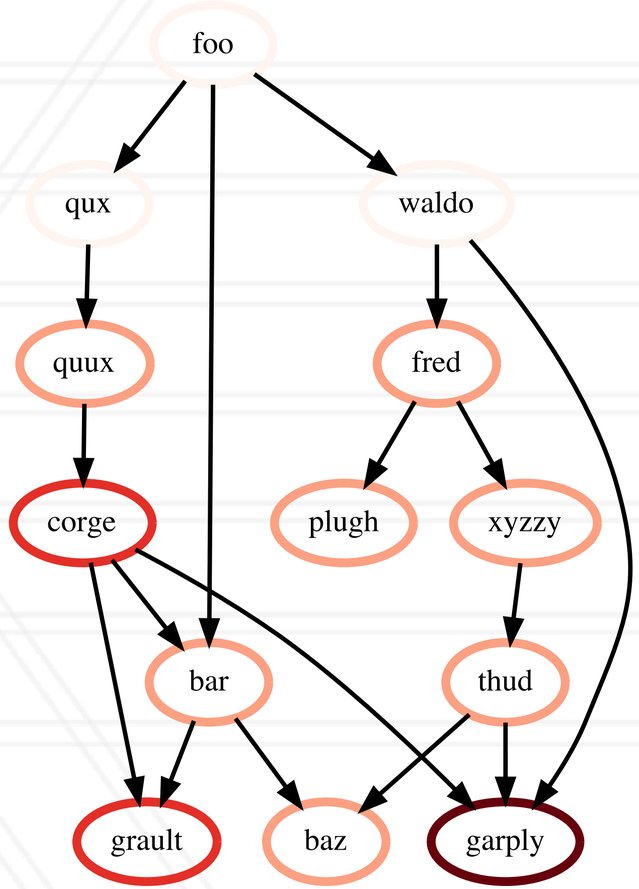
Calling context tree (CCT)

Contextual information

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

Performance Metrics

- Time
- Flops
- Cache misses



Call graph

Questions?

UNIVERSITY OF
MARYLAND

Abhinav Bhatele

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

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