



SEJITS Group Aspire Lab U.C. Berkeley

Chick Markley
chick@berkeley.edu
Staff Researcher

Efficiency Layer



Performance Gap

| | | | |
|------------------|---------------------|-----------------|----------------|
| Graph Processing | Multimedia Analysis | Computer Vision | Software Radio |
|------------------|---------------------|-----------------|----------------|

languages (Python, Scala)

C/assembler, CUDA/OpenCL, Java)



- Productivity programmers work on problem in application domains
- Efficiency programmers focus on frameworks and libraries for
- What kinds of frameworks and efficiency layer provide?

SEJITS

- Selective
- Embedded
- Just-In-Time
- Specializers

Pattern Specific Optimization

Patterns

- Stencil
- Linear Algebra
- Reduction
- Support Vector Machine

Tools

- Python
- C/C++
- OpenCL
- OpenMP
- Chisel, FPGA
- LLVM

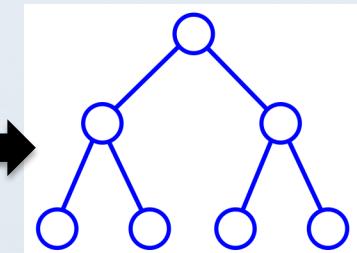
Anatomy of SEJITS

ctree

specializer

ctree

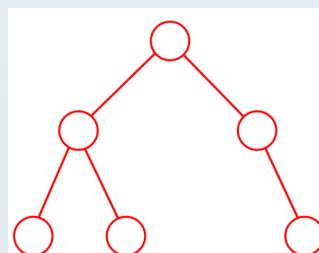
```
while x:  
    x += y  
z = x / y  
. . .  
. . .
```



DSL program
embedded in
Python

Python AST

Semantic Model



void *fn

C AST

OpenTuner Integration

- Originally homegrown auto-tuning
- Initial integration with OpenTuner
- Using refactored opentuner api
 - Opentuner/examples/py_api.py

Parameters

- Relatively small ranges
- Small number of parameters

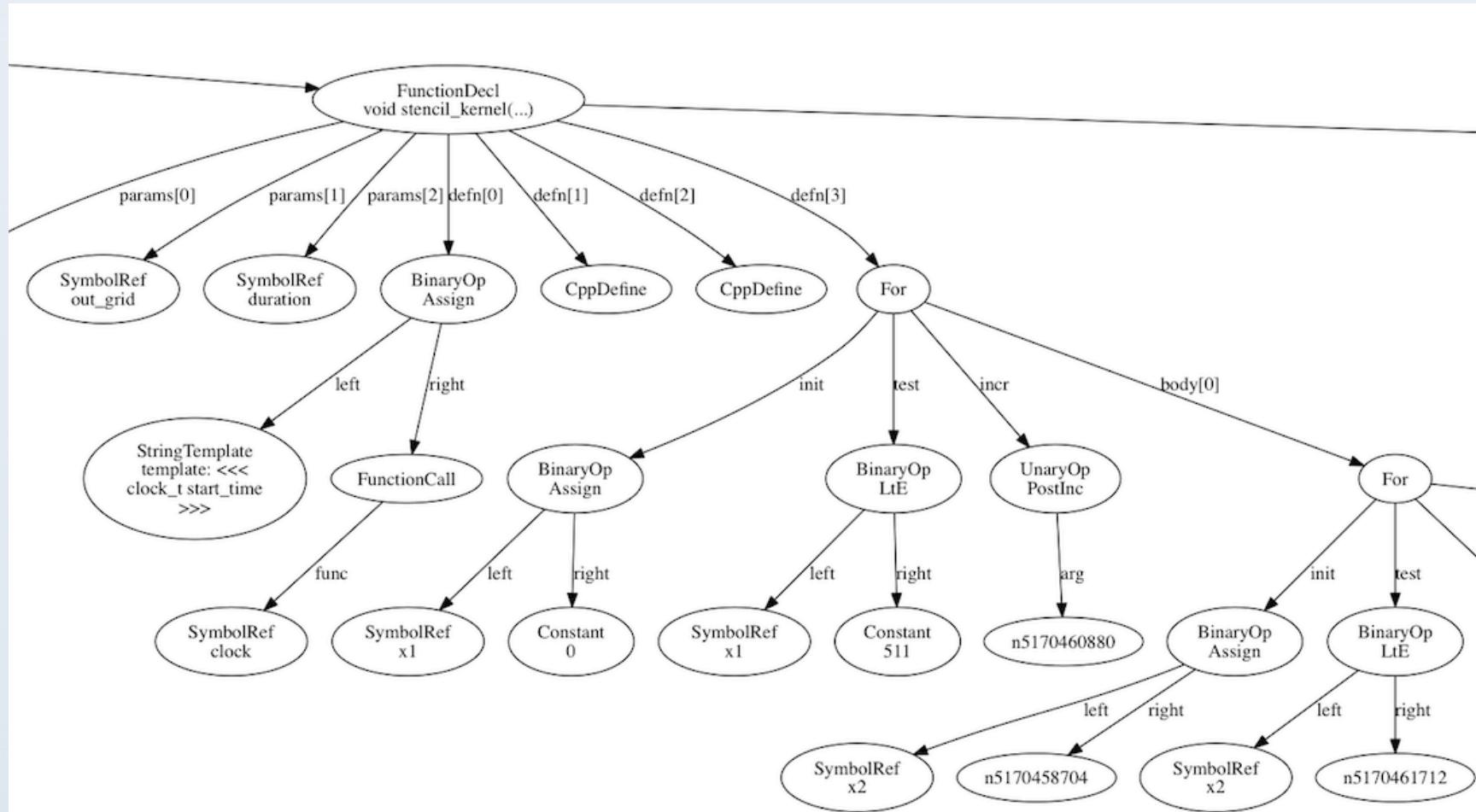
Stencil example

```
manip = ConfigurationManipulator()
manip.add_parameter(PowerOfTwoParameter("unroll_factor", 1, 4))
manip.add_parameter(PowerOfTwoParameter("block_factor1", 4, 8))
manip.add_parameter(PowerOfTwoParameter("block_factor2", 4, 8))
manip.add_parameter(PowerOfTwoParameter("block_factor3", 4, 8))
# hack to induce parameter retry
manip.add_parameter(IntegerParameter("x", -10, 10))
```

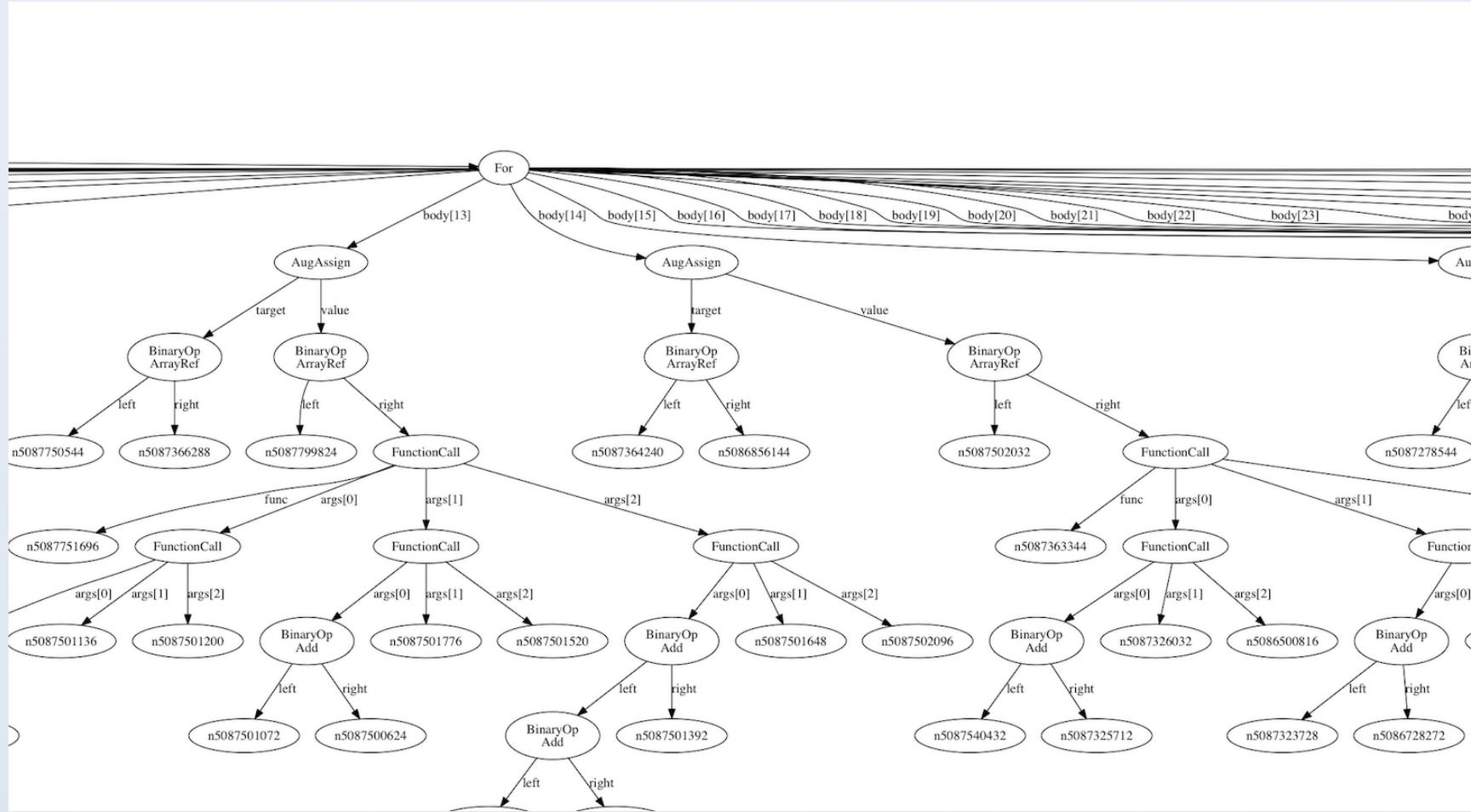
Drives AST transformation

```
first_For = tree.find(For)
inner_For = Optimizer.FindInnerMostLoop().find(first_For)
tree = optimizer.unroll(tree, inner_For, unroll_factor)
inner, first = optimizer.block_loops(
    inner_For,
    first_For,
    block_factors + [1])
```

Before 13600 × 1697 pixels



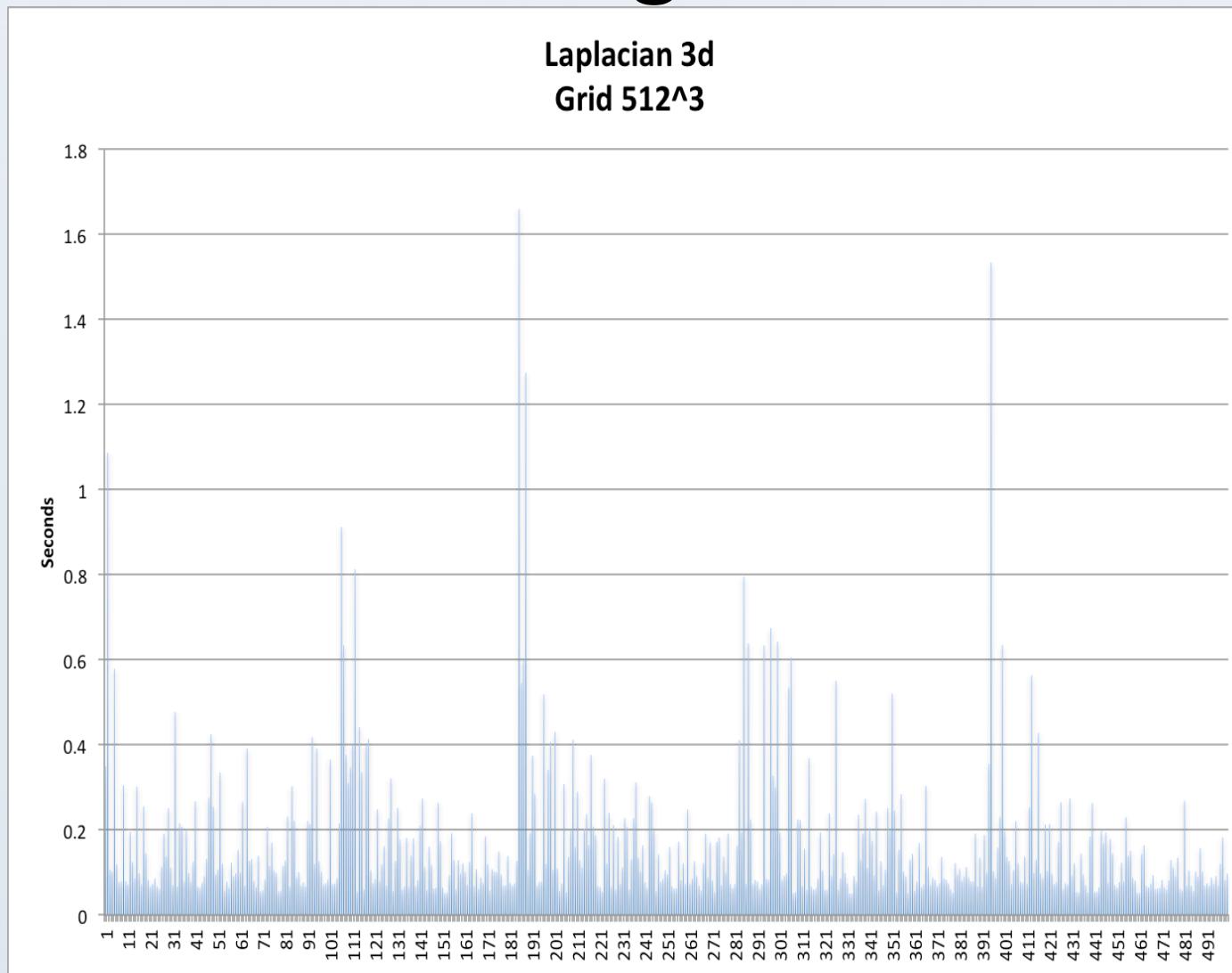
After 46824×2215 pixels



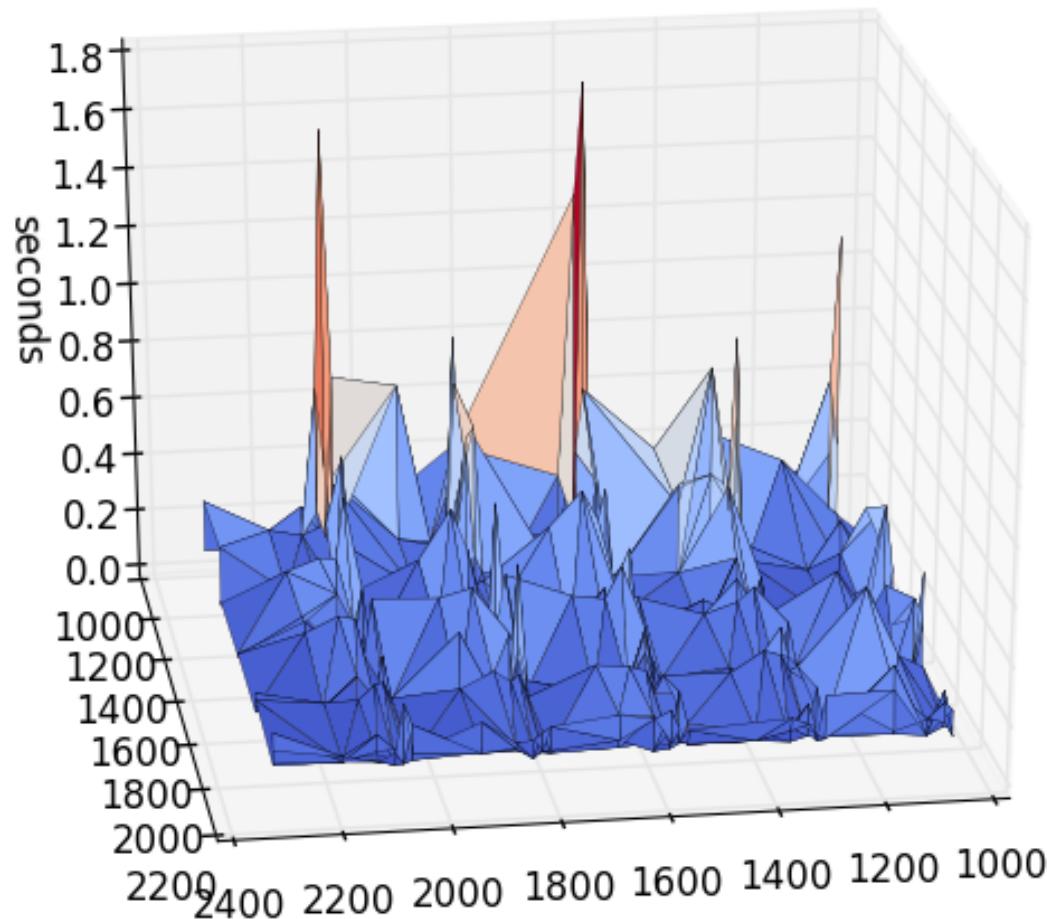
Sample Run

- Intel E5-2667v2, 3.30GHz
- Laplacian 3d Grid size 1028x128x128
- 500 trials
- Best time 0.0506
- Average time 0.1637
- Worst time 1.6589
- Best is 3x versus average, 30x better than worst

Tuning Run



Laplacian 3-d
Grid size 512, 256, 128



Current Work

- Hindemith
- HPGMG
- Chisel FPGA backend

OpenTuner Todo

- Tuning space relationship to work sizes
- Threading issues
- Small search spaces
- Hierarchical search spaces
- Composition of tuning parameters across specializers

Thanks

Wish list

- Hierarchical
 - Gpu vs. MIC
 - Compiler
 - Compiler flags
- Composition & Fusion
 - DSL
 - compilers