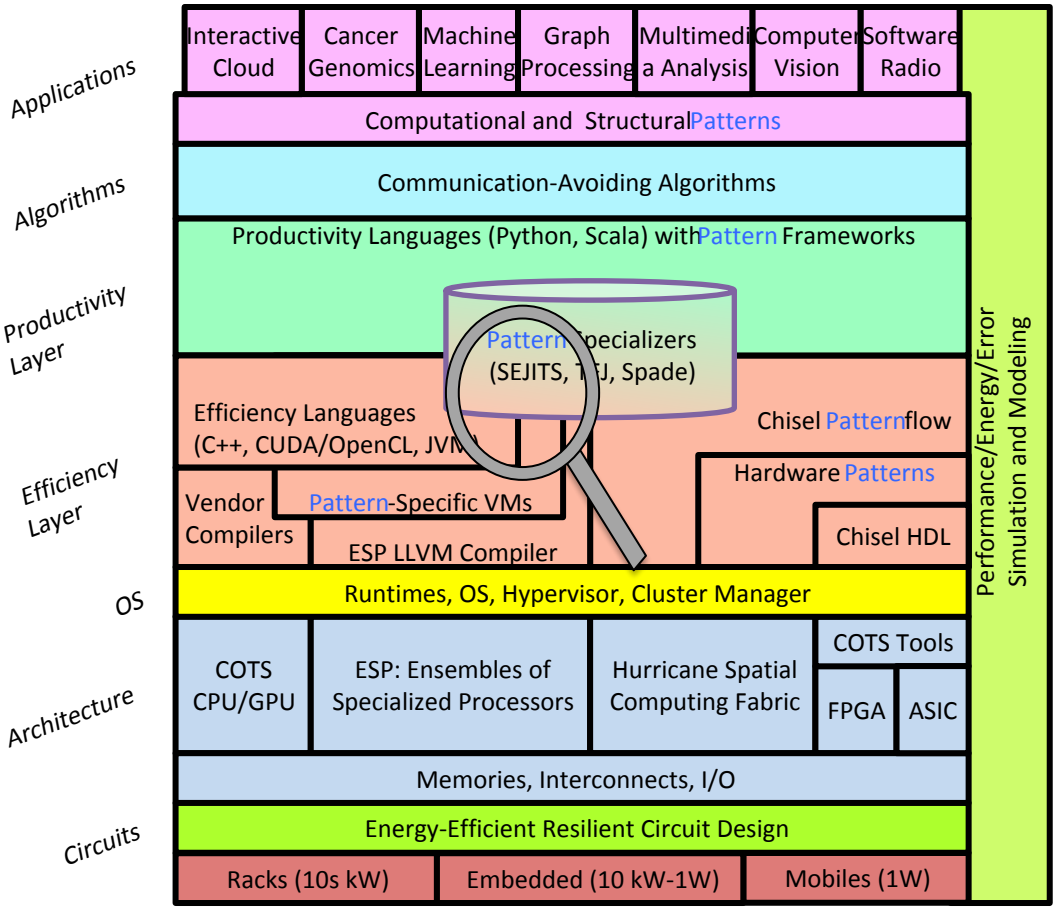
SEJITS
Chick Markley: Staff Programmer

- Selected
- Embedded
- Just-In-Time
- Specialization



SEJITS:

It's a methodology

- Write in a high level language
- Identify performance bottle neck
- Use JIT code generation to optimize
- Used in the projects of numerous graduate students

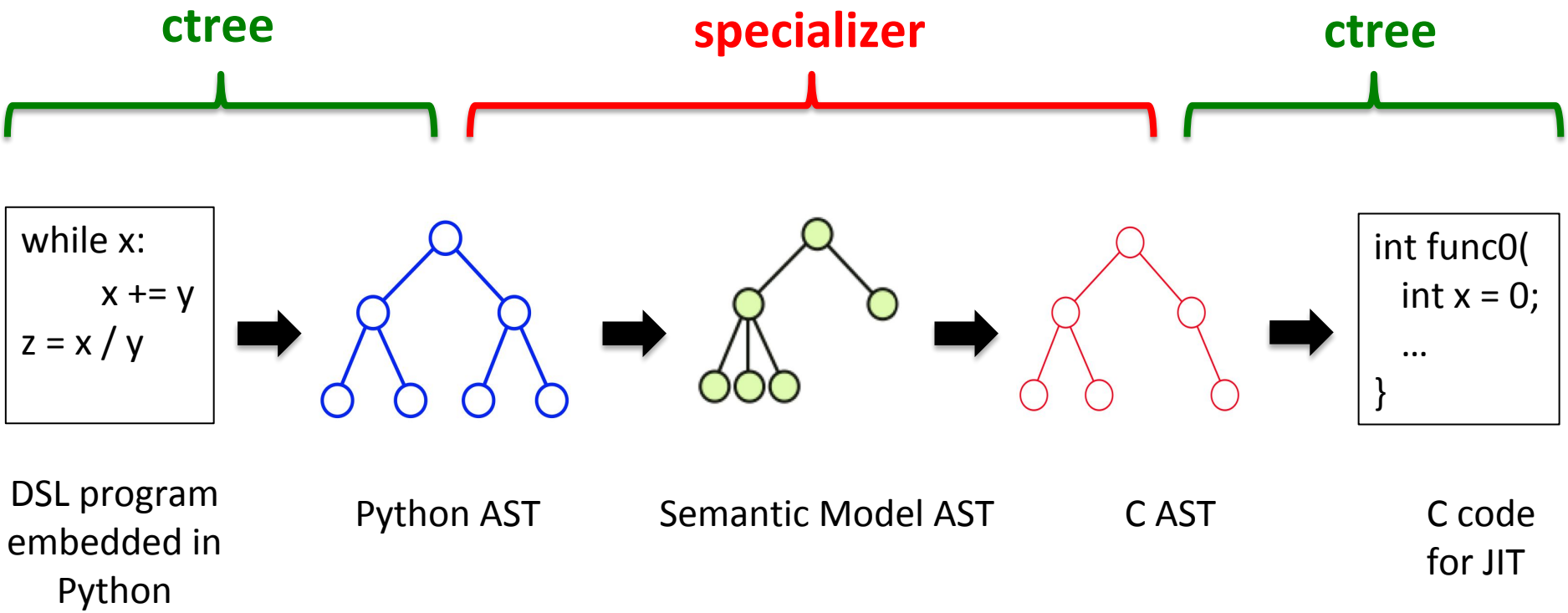
Implemented ASP Specializers

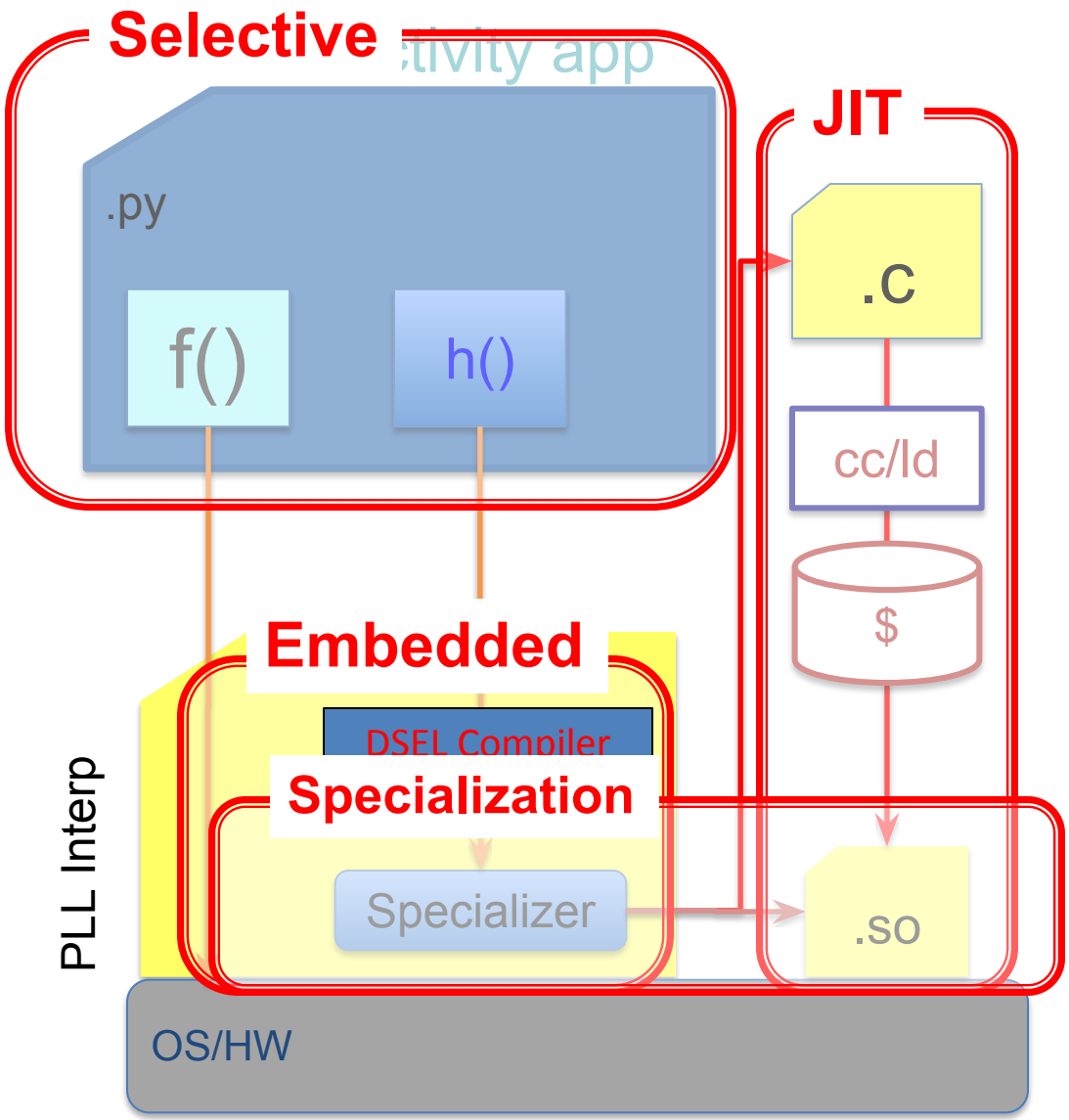
DSEL/Library	Platforms
Stencil/Structured Grid (Shoaib Kamil)	x86+OpenMP
Semantic Graphs Filtering & Semiring Operations in KDT (Aydin Buluc)	x86+MPI
Parallel Map (Michael Driscoll)	x86+processes, cloud
PyCASP/Gaussian Mixed Models/GMM (Katya Gonina)	CUDA, Cilk Plus
CA Matrix Powers for CA Krylov Subspace Methods (Jeffrey Morlan)	x86+pthreads
Bag of Little Bootstraps (Peter Birsinger, Richard Xia, Aakash Prasad)	x86+Cilk Plus, Cloud via Spark
Map Over Combinations (Shoaib Kamil)	Python

SEJITS:

- It's a software framework
  - Let's build on what's gone before.
  - Ctree implement SEJITS as a software library
  - Integrate with autotuning.
  - Most of the projects implemented by undergraduates

Anatomy of specialization




Selective
.py
f()
h()
JIT
.c
cc/ld
$
Embedded
DSEL Compiler
Specialization
Specializer
.so
PLL Interp
OS/HW

Implemented ASP Specializers

DSEL/Library	Platforms
Support Vector Machine (Michael Lin)	x86
A* Search (Hugo Barreto)	x86
Latte (Full Python Caffe (Leonard Truong)	x86 + GPU
Skye (real-time movement tracking (Mihir Patil)	X86 + OpenCV
Generating Chisel from stencils (Sean Roberts)	Python
Magnetic Particle Imaging (Mihir Patil)	X86, GPU
PyGMG: Snowflake (Nathan Zhang, Shiv Sundram)	X86, MPI, GPU

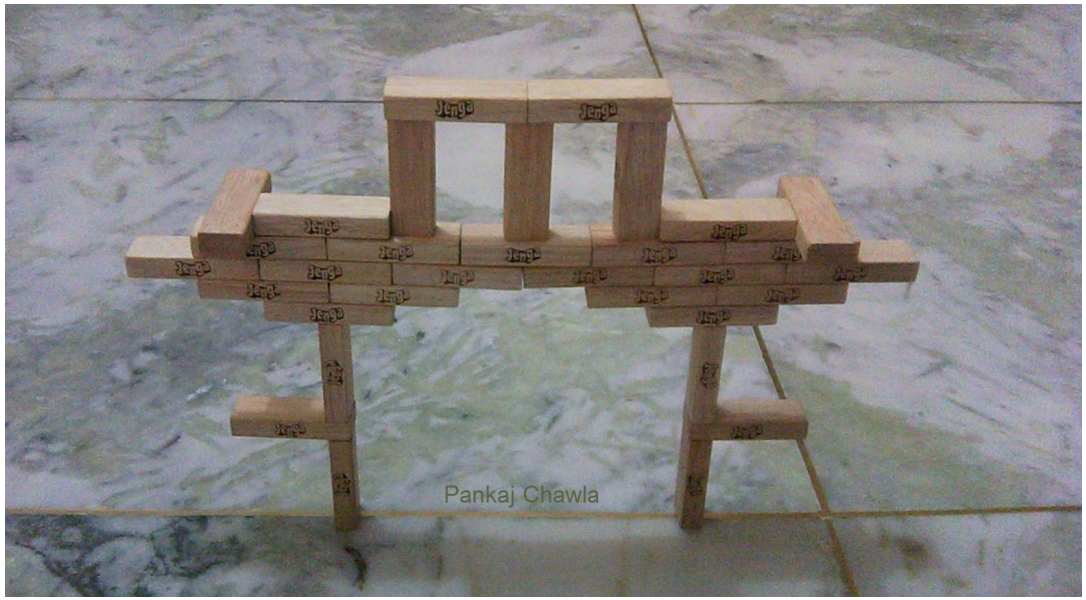
`

Snowflake & PyGMG

Lessons Learned

- As time went on we had fewer and fewer graduate students
- The SEJITS team moved to staff and more undergraduates
- Pattern coverage increased
- New deep results decreased

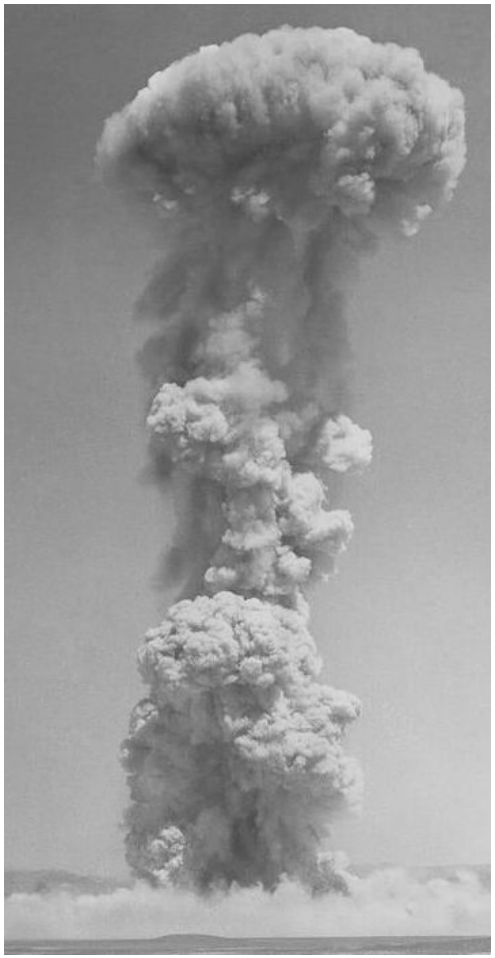
Solid underpinnings



- SEJITS relied on many external libraries.
- Some were small and poorly supported
- Bugs in libraries forced upgrades
- Upgrades led to incompatibilities.
- Changing horses required re-engineering

Finding Good Applications

- Sometimes close is good enough
- When 2 or 3 order of magnitudes improvements are achieved easily
- Somehow the effort to get another few percent is harder to find



Writing specializers

- The skill is orthogonal to the skills of performance and productivity programming
- Even in selective scopes it’s still writing compiler.
- Hard to teach, hard to learn.

SEJITS, not just performance library,
it's also a really good sieve

Thank you

Thank you.

SEJITS:

"Selective, Embedded, Just-In Time Specialization"

- SEJITS bridges productivity and efficiency layers through specializers embedded in a modern high-level productivity language, Python
  - Embedded "specializers" use language facilities to map high-level patterns to efficient low-level code (at run time, install time, or development time)
  - Specializers can incorporate or package auto-tuners

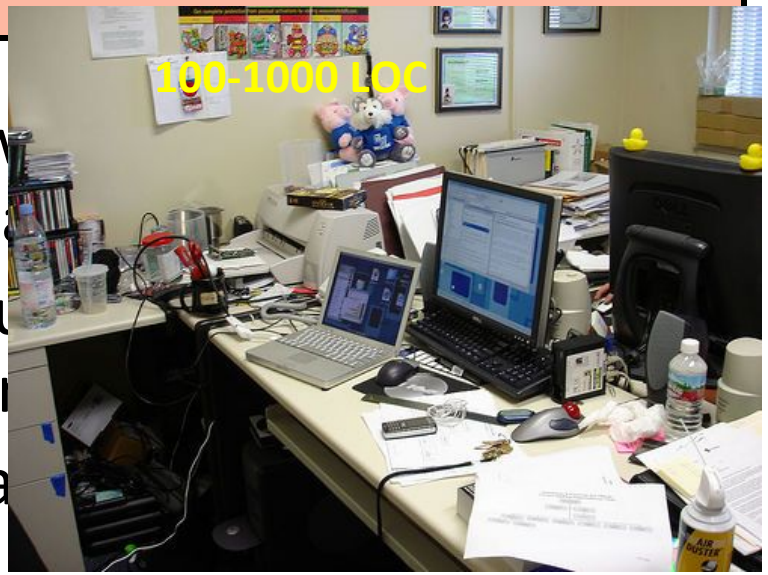
Productivity-Performance Gap

Effici
Laye

Graph rocessing	Multimedia Analysis	Computer Vision	Software Radio

guages (Python, Scala)

C/assembler, CUDA/OpenCL, Java)



- Productivity programmers w
  problem in application doma
- Efficiency programmers focu
  frameworks and libraries for
- What kinds of frameworks a
  efficiency layer provide?