Parallel Scientific, Inc.

Peter Braam – Founder / Chief Architect
Q2 2012

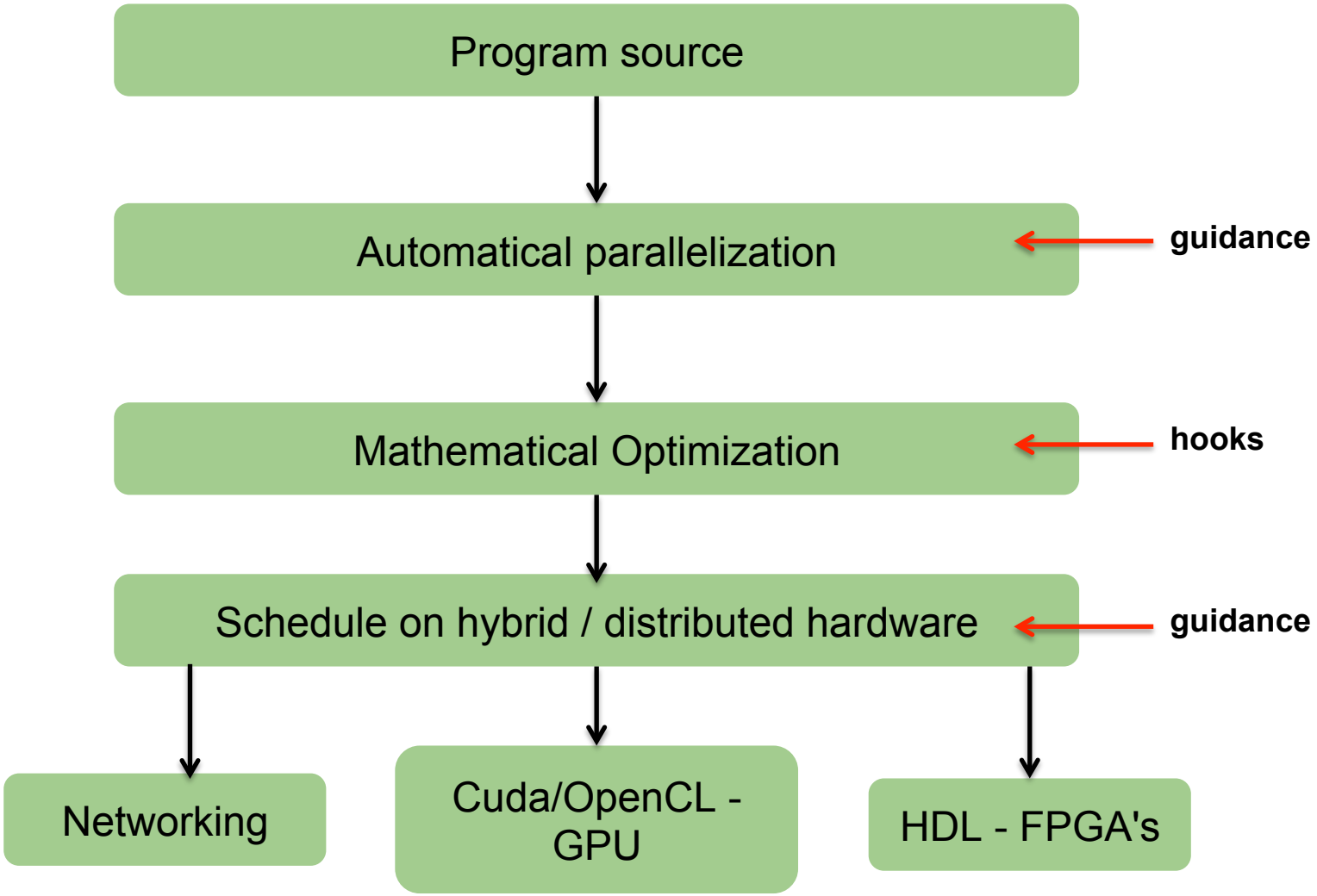
Parallel Scientific:10 amazing minutes

Founded 4/2010, slow start
Now 10 tech-staff, 50% in US
Some funding

Goal - Dramatic Improvements in some aspects of parallel programming. Products & contract work.

Expertise - programming languages, hardware description, mathematics

Proposed Parallel Computing Stack



Automatic Parallelization

Sparse dot product

Sequential:

```
svMul :: [ (Int,Float) ] -> [ Float ] -> Float
svMul sv w = sum [ v * (w ! i) | (v, i) <- sv ]
```

A sparse vector is a list of non-zero pairs

The i-th element of w

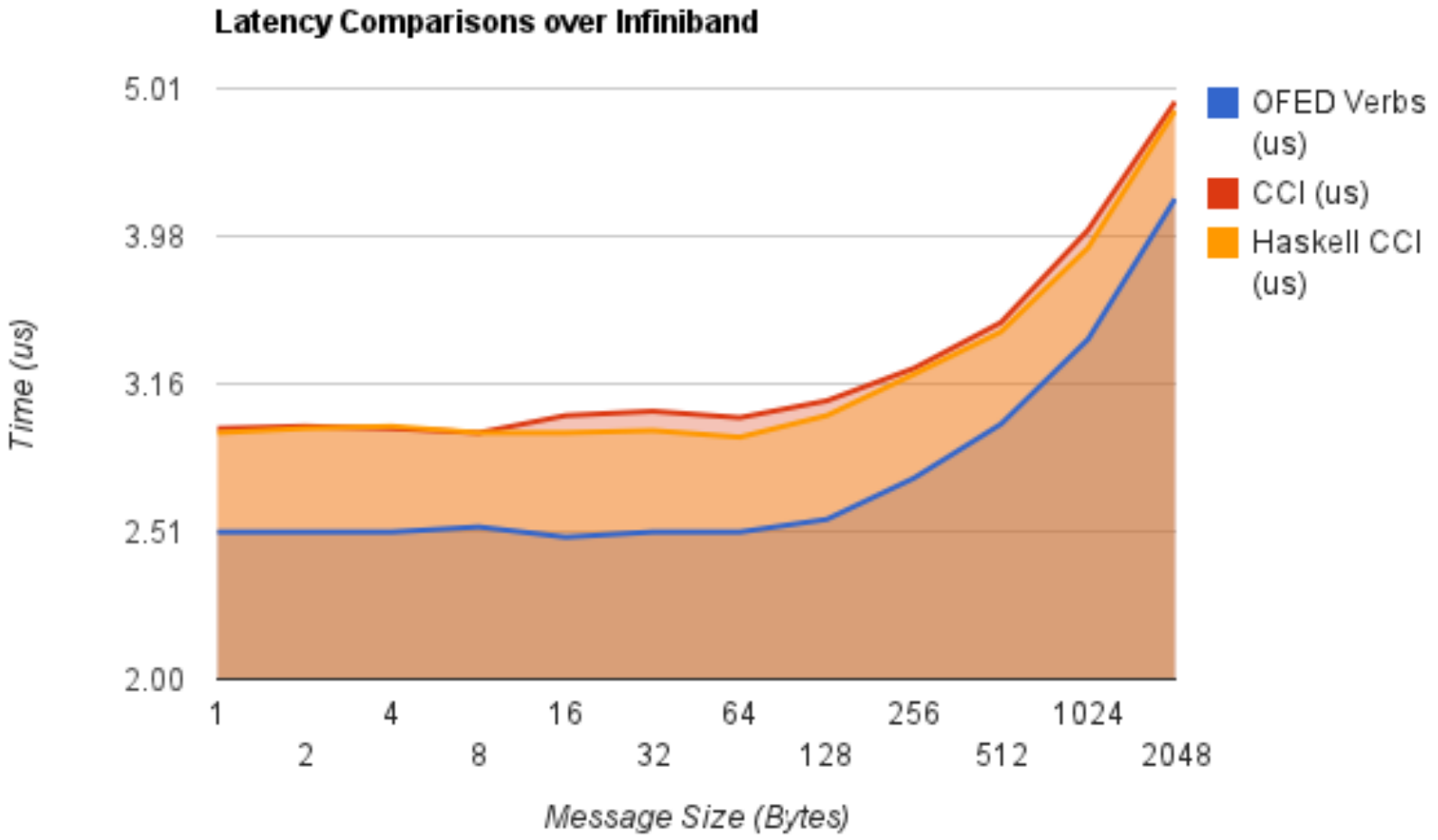
Array comprehension – select (v,i) from sv

Parallel – use parallel arrays [: …. :]

```
svMul :: [: (Int,Float) :] -> [: Float :] -> Float
svMul sv w = sum [: v * (w ! i) | (v, i) <- sv :]
```

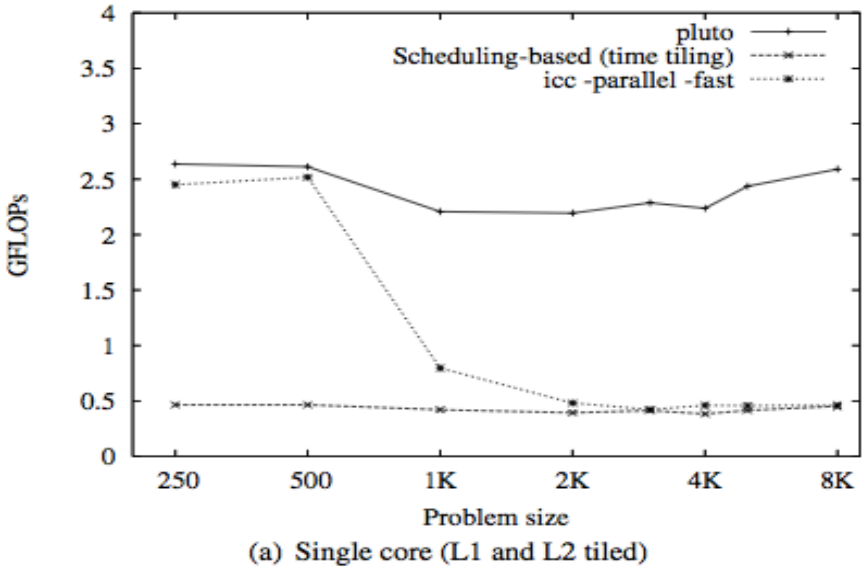
Our executable business card …

Performance - Infiniband

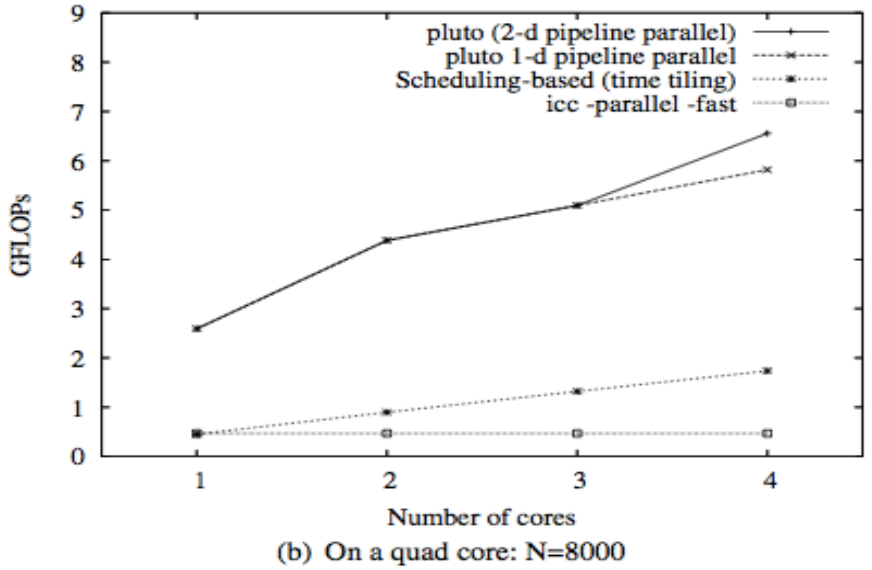


Mathematical Code Optimization

- PLUTO – from Ohio / Louisianna
- Mathematical optimization gets further than icc
- Haskell HOOPL interface will do this



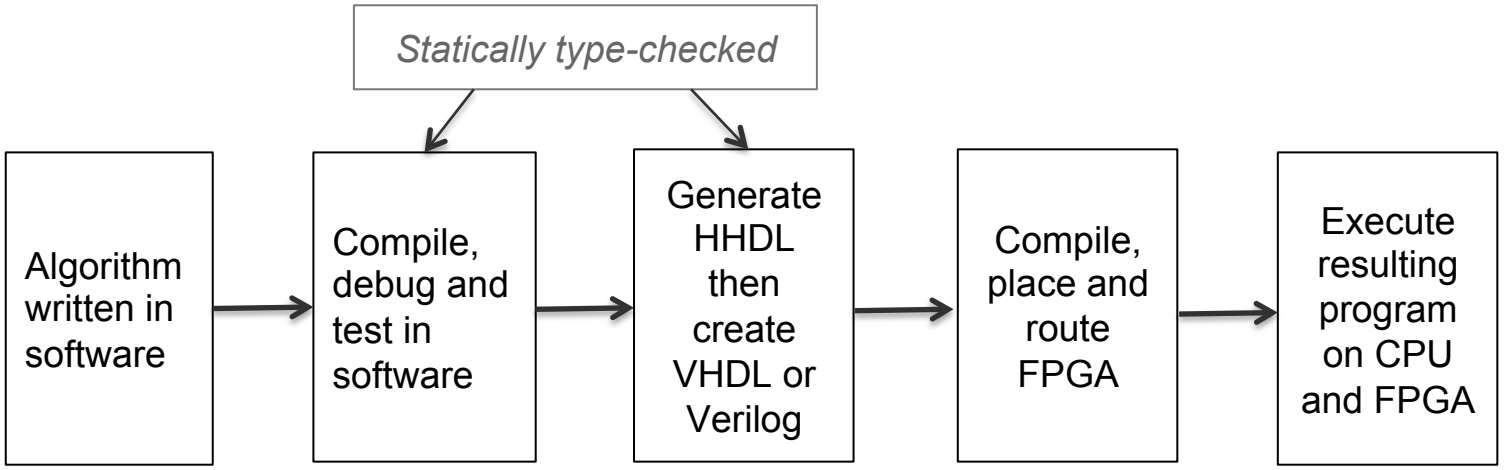
(a) Single core (L1 and L2 tiled)



(b) On a quad core: N=8000

Figure 10. LU performance

FPGA & Hybrid development



- FPGA or Hybrid Program Design
  - One program

Make data intelligence faster

- Arista FPGA switch 7124FX
  - 24 port 10Gige Switch with large Altera FPGA
  - FPGA has 160Gbps throughput and 8 ports
- Opportunities
  - Line rate data filtering / analytics in the switch
  - Reduced problem sent to CPU

GraphHammer: Example cont'd

Domain Specific Language – streaming graph analytics

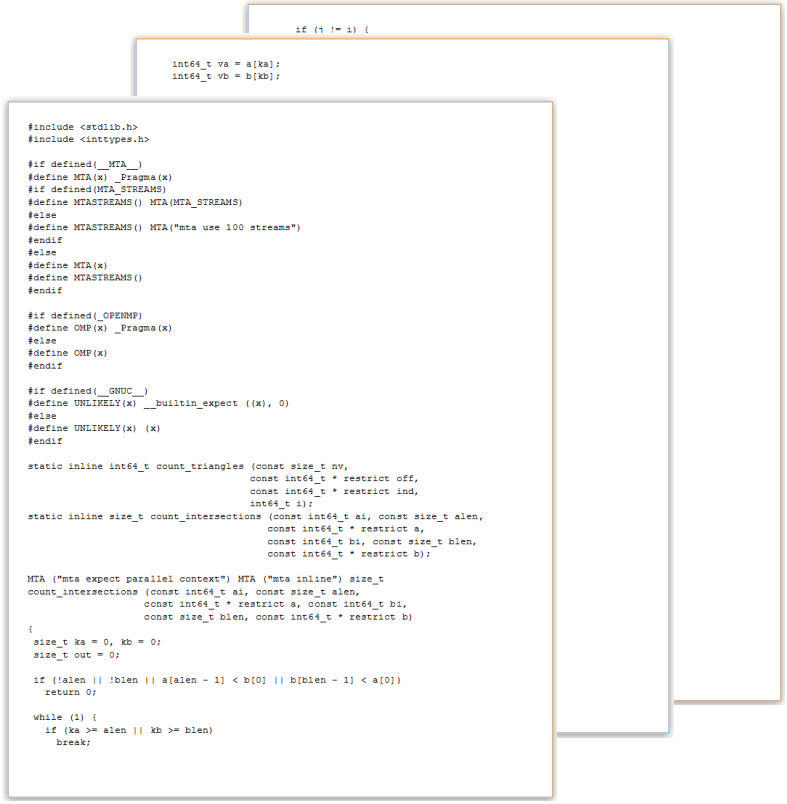
Inspired by David Bader's group – STINGER

Comparable performance – much less code

Compiles to anything, no explicit parallelism

```
int count;

triangleCount (src,dst) {
  int n = 0;

  for x in edgesFrom src
    for y in edgesFrom dst
      if x == y then
        (count x) := (count x) + 1;
        n := n + 1;
  (count src) := (count src) + n;
  (count dst) := (count dst) + n;
}
```

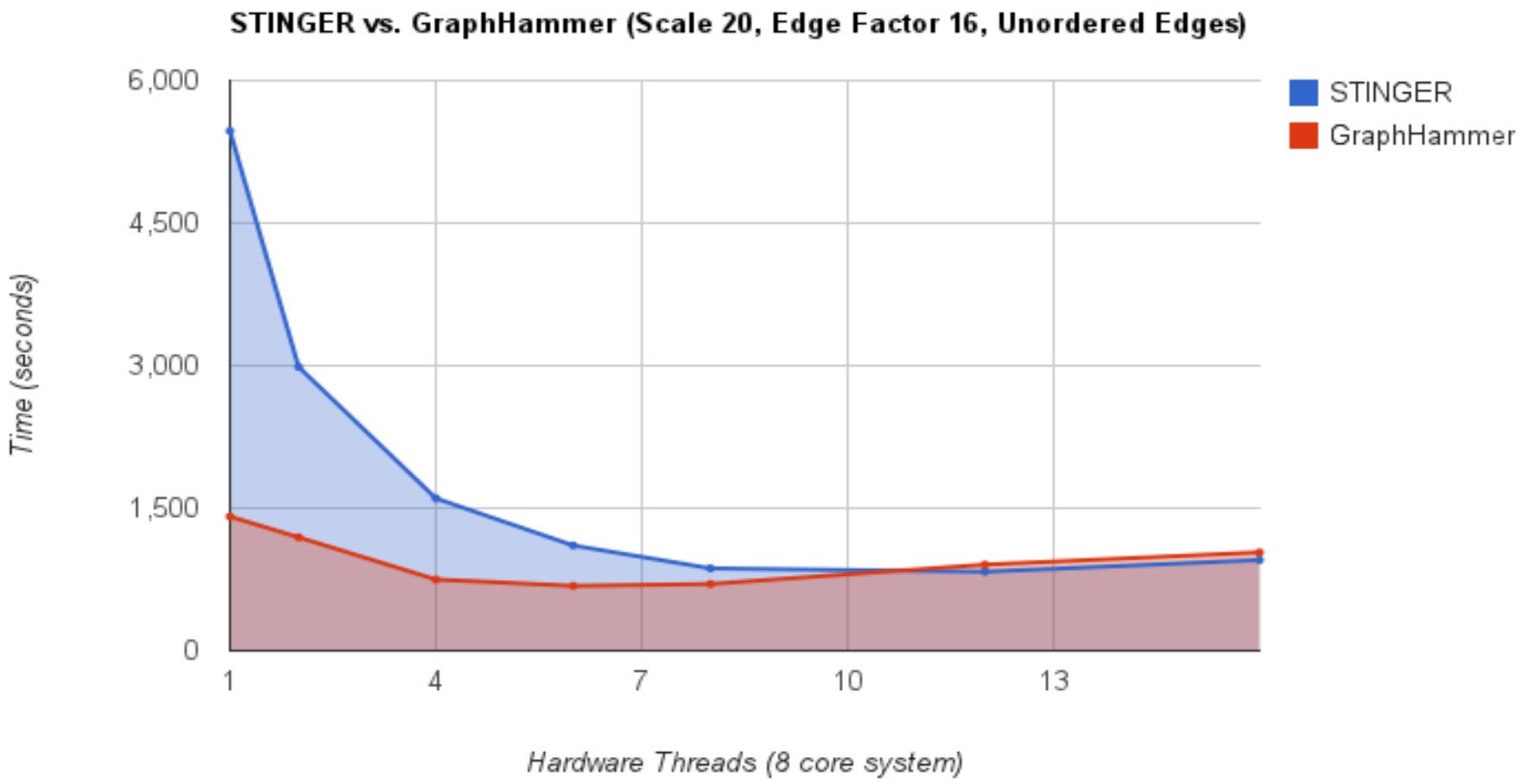
GraphHammer



```
if (i != i) {

int64_t va = a[ka];
int64_t vb = b[kb];

#include <stdlib.h>
#include <inttypes.h>

#if defined(__MTA__)
#define MTA(x) _Pragma(x)
#if defined(MTA_STREAMS)
#define MTASTREAMS() MTA(MTA_STREAMS)
#else
#define MTASTREAMS() MTA("mta use 100 streams")
#endif
#else
#define MTA(x)
#define MTASTREAMS()
#endif

#if defined(_OPENMP)
#define OMP(x) _Pragma(x)
#else
#define OMP(x)
#endif

#if defined(__GNUC__)
#define UNLIKELY(x) __builtin_expect ((x), 0)
#else
#define UNLIKELY(x) (x)
#endif

static inline int64_t count_triangles (const size_t nv,
                                       const int64_t * restrict off,
                                       const int64_t * restrict ind,
                                       int64_t i);
static inline size_t count_intersections (const int64_t ai, const size_t alen,
                                          const int64_t * restrict a,
                                          const int64_t bi, const size_t blen,
                                          const int64_t * restrict b);

MTA ("mta expect parallel context") MTA ("mta inline") size_t
count_intersections (const int64_t ai, const size_t alen,
                     const int64_t * restrict a, const int64_t bi,
                     const size_t blen, const int64_t * restrict b)
{
 size_t ka = 0, kb = 0;
 size_t out = 0;

 if (!alen || !blen || a[alen - 1] < b[0] || b[blen - 1] < a[0])
   return 0;

 while (1) {
   if (ka >= alen || kb >= blen)
     break;
```

Hand-crafted C

GraphHammer Performance

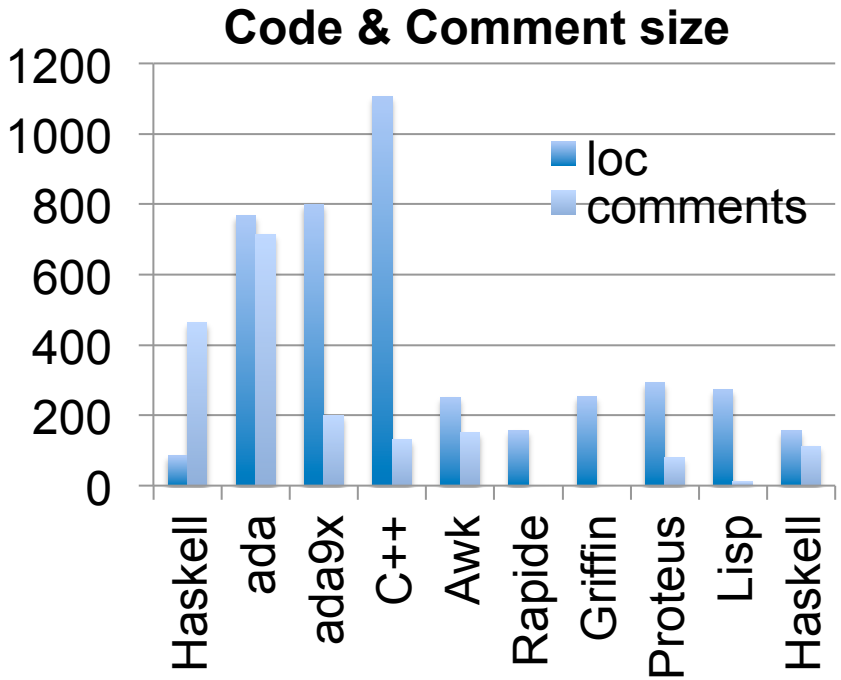


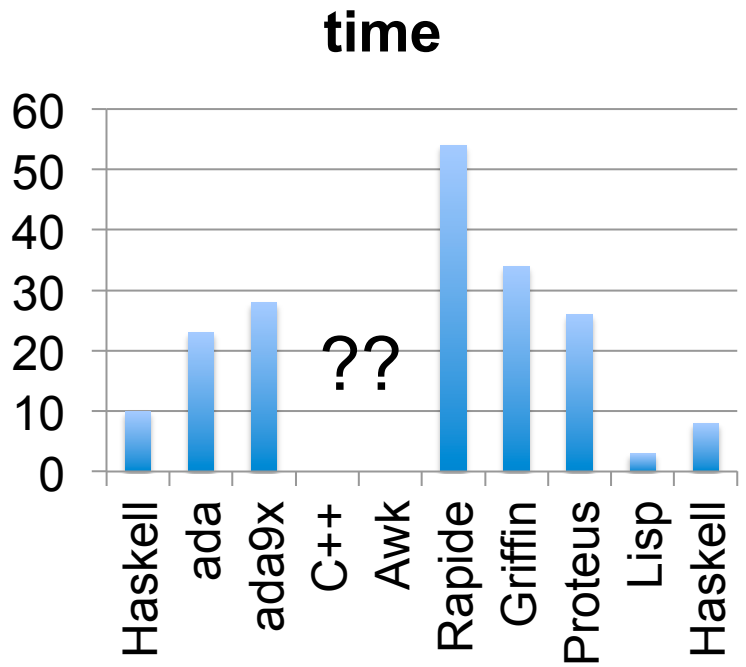
Haskell Productivity

Every bank has a “secret” Haskell group

Much commercial success in hardware design

Amazing open source community





Thank you!

Questions?

peter.braam@parsci.com

info@parsci.com
+1 650.515.4523