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

1. Program source

2. Automatical parallelization

3. Mathematical Optimization

4. Schedule on hybrid / distributed hardware

5. Networking

6. Cuda/OpenCL - GPU

7. HDL - FPGA's
Sparse dot product

**Sequential:**

$$\text{svMul :: [ (Int,Float) ] -> [ Float ] -> Float}$$

$$\text{svMul sv w = sum [ v * (w ! i) | (v, i) <- sv ]}$$

**Parallel** – use parallel arrays [ : ..... : ]

$$\text{svMul :: [:(Int,Float):] -> [: Float : ] -> Float}$$

$$\text{svMul sv w = sum [: v * (w ! i) | (v, i) <- sv :]}$$

A sparse vector is a list of non-zero pairs

The i-th element of w

Array comprehension – select (v,i) from sv
Our executable business card ...
Performance - Infiniband

Latency Comparisons over Infiniband

Time (us)

Message Size (Bytes)

OFED Verbs (us)
CCI (us)
Haskell CCI (us)
Mathematical Code Optimization

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

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

```c
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 Performance

STINGER vs. GraphHammer (Scale 20, Edge Factor 16, Unordered Edges)

Time (seconds)

Hardware Threads (8 core system)
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