Beauty And The Beast
Exploiting GPUs In Haskell

Alex Cole
University of Leicester

CPA 2012, Dundee
27\textsuperscript{th} August 2012

Alistair A McEwan
University of Leicester

Geoff Mainland
Microsoft Research
(x + a)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k}
[Only] a few of the proposed approaches [...] are currently used in practice due to the unavailability of the software or due to difficulties with incorporating the new tools in the existing data analysis pipelines.

- Nesvizhskii10
Hypotheses

• Haskell is easy
• But performance (lists) stinks
• Using a DSL we can recover some “C” performance
• We think the trade-off is acceptable
• Based on this, we can address the richer MS computation domain with ease and performance.
Whole Array
/ Index

- Shift .<<<
- Add .+
- Multiply .*

Per-Element

- Shift ^<<<
- Add ^+
- Multiply ^*
2 4 5 7 3 9

* 1 3 1

= 10 19 18 29 25 30
-- Sum up all the parts for the final answer.
convolve filter input = \text{sum convolvedParts}

where

-- Convolve the whole array with one shift and multiply.
\text{oneFilterElement shift mul} = (\text{input shift mul})

-- Get the filter radius.
r = \text{length filter `div` 2}

-- Do all the shifts and multiplies separately.
\text{convolvedParts} = \text{zipWith oneFilterElement} \ [-r..r] \text{ filter}
• Using Lists:

\[
\text{convolve} :: [\text{Float}] \rightarrow [\text{Float}] \rightarrow [\text{Float}]
\]

• Using PEGGY:

\[
\text{convolve} :: [\text{Float}] \rightarrow \text{PYExpr Float target} \rightarrow \text{PYExpr Float target}
\]

-- Convert the old input to PEGGY GPU data.
peggyInput = toArray listInput
data PYExpr a b = (PYCode a b) (PYMem a b)

data instance PYMem a TargetHaskell = [a]

instance PYCode a TargetHaskell where
  (^+) x y = zipWith (+) x y
STOP GLOBAL WARMING: BECOME A PIRATE

Global Average Temperature Vs. Number of Pirates

WWW.VEGANZA.ORG
DATA

* f1 =  

LOW FREQUENCY

* f =  

HIGH FREQUENCY

* f =  

LOW FREQUENCY

* f =  

HIGH FREQUENCY
HIGH FREQUENCY
CUT OFF FILTER
DATA
The Future
NEXT EXIT
Overview

• Backend with proven potential

• Complete MS analysis implementation

• Abstract to a DSL

• Improved performance
Nesvizhskii10: