Haskell in Green Land: Analyzing the Energy Behavior of a Purely Functional Language

Luís Gabriel Lima[1], Gilberto Melfe[2], Francisco Soares-Neto[1], Paulo Lieuthier[1], João Paulo Fernandes[2], and Fernando Castor[1]

[1] {lgnfl, fmssn, pvjl, castor}@cin.ufpe.br
[2] gilbertomelfe@gmail.com, jpf@di.ubi.pt

Global energy system is unsustainable

Bloomberg the Company & Its Products | Bloomberg Anywhere Remote Login | Bloomberg Terminal Demo Request

Bloomberg Business | News | Markets | Insights | Video

Technology

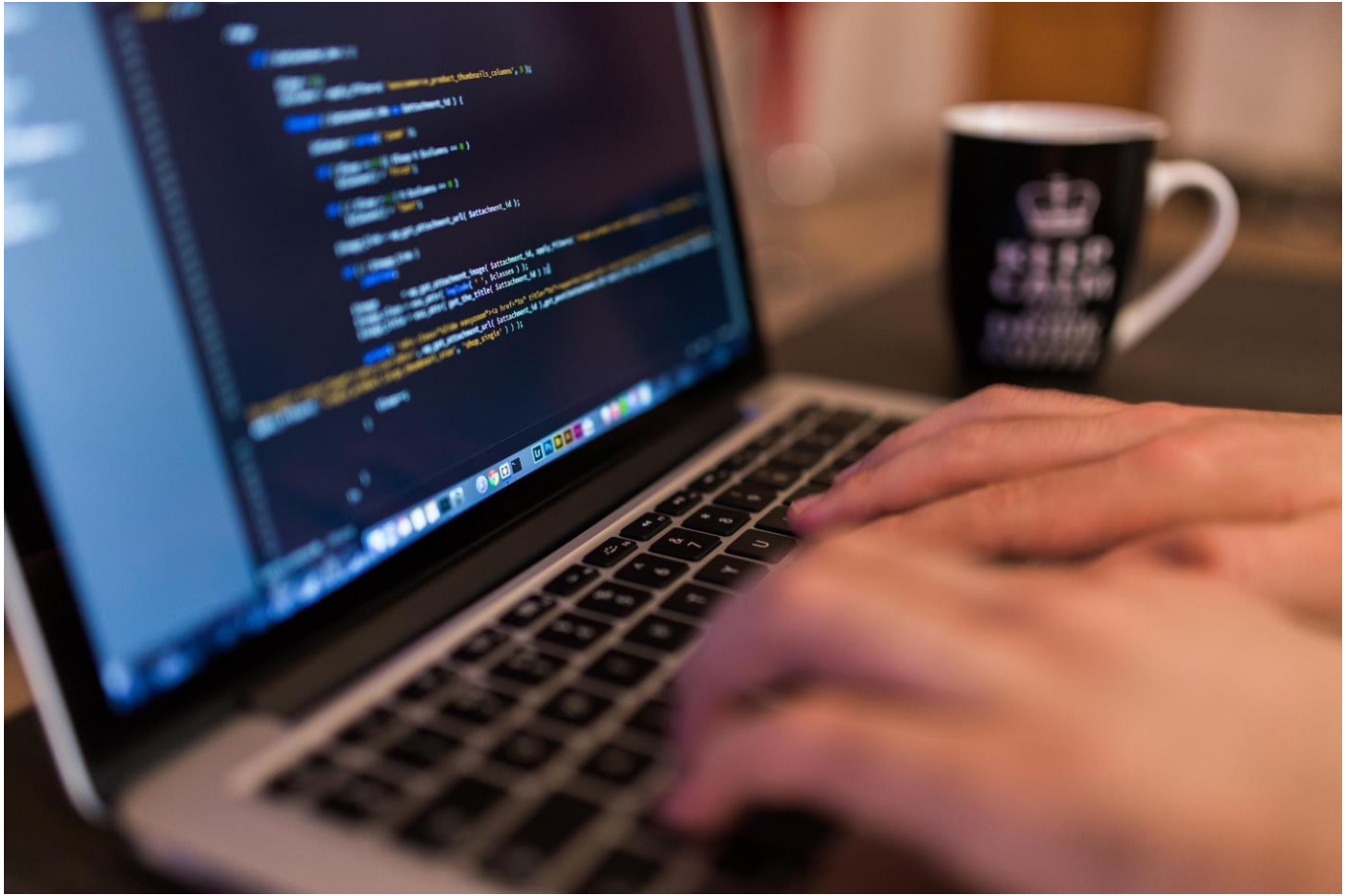
Inside the Arctic Circle, Where Your Facebook Data Lives

By Ashlee Vance | October 04, 2013

SEND TO kindle

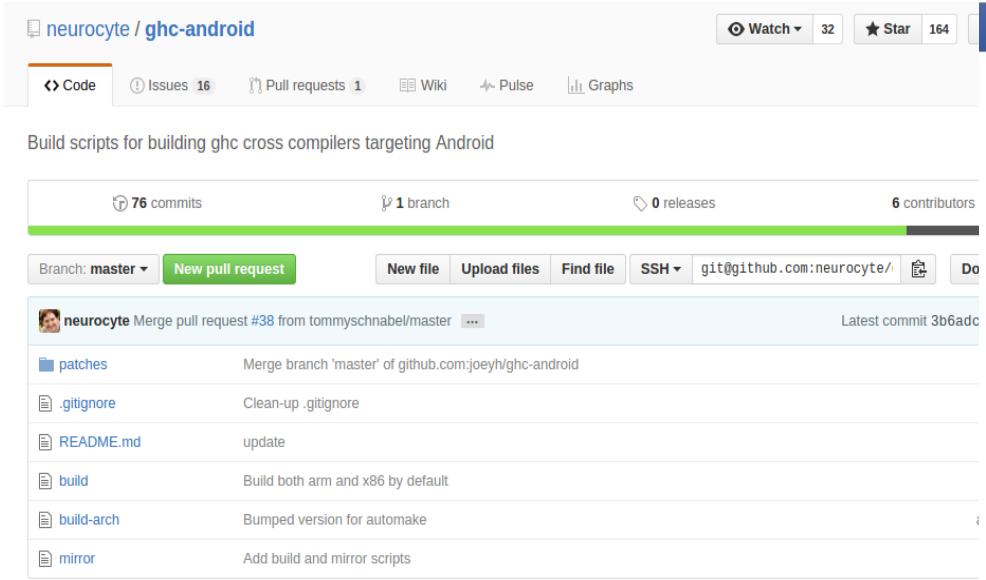
Photograph by Jasper Doest/Foto Natura/Minden Pictures/Corbis

Every year, computing giants including Hewlett-Packard (HPQ), Dell (DELL), and Cisco Systems (CSCO) sell north of $100 billion in hardware. That's the total for the basic iron—




LAMBDA LABS
LAMBDA LOUNGE

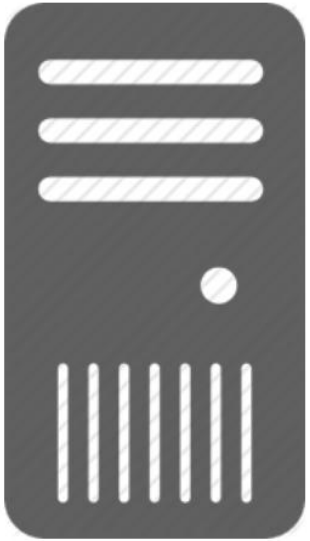
Haskell has feelings too!



The adventures of Haskell in Greenland

To what extent can we save energy by refactoring existing Haskell **programs** to use different **data structure implementations** or **concurrent programming constructs**?

Experimental Setup



2x10-core Intel Xeon E5-2660 v2 processors (Ivy Bridge) 256GB of DDR3 1600MHz RAM

Criterion

RAPL

Study 1: Purely functional data structures

RQ1. How do **different implementations** of the same **abstractions** compare in terms of **run time** and **energy efficiency**?

RQ2. For concrete operations, what is the **relationship** between their **performance** and their **energy consumption**?

Study 1: Edison Library

Collections	Associative Collections	Sequences
EnumSet StandardSet UnbalancedSet LazyPairingHeap LeftistHeap MinHeap SkewHeap SplayHeap	AssocList PatriciaLoMap StandardMap TernaryTrie	BankersQueue SimpleQueue BinaryRandList JoinList RandList BraunSeq FingerSeq ListSeq RevSeq SizedSeq MyersStack

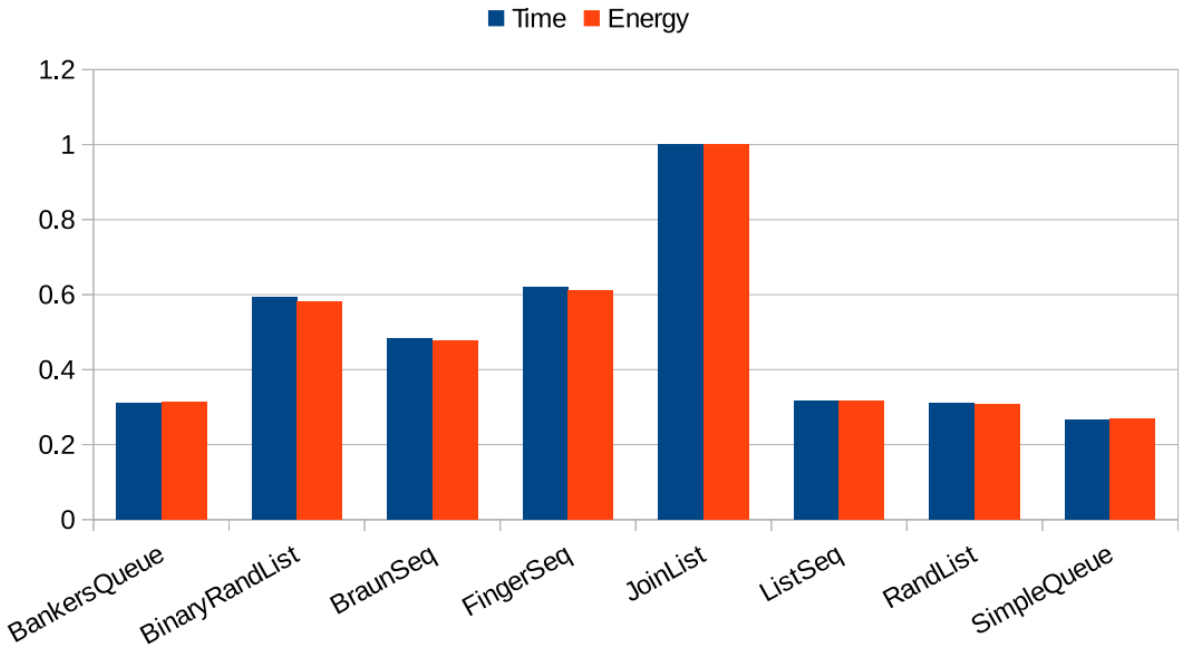
Study 1: Benchmark

iters	operation	base	aux
1	add	100000	100000
1000	addAll	100000	1000
1	clear	100000	n.a.
1000	contains	100000	1
5000	containsAll	100000	1000
1	iterator	100000	n.a.
10000	remove	100000	1
10	removeAll	100000	1000
5000	toArray	100000	n.a.
10	retainAll	100000	1000

```
iters = 0;
while iters < 10
retainAll base aux;
iters++;
```

320 configurations
3000+ executions

Study 1: Results



RQ1. How do **different implementations** of the same **abstractions** compare in terms of **runtime** and **energy efficiency**?

Full details on **green-haskell.github.io**.

RQ2. For concrete operations, what is the **relationship** between their **performance** and their **energy consumption**?

Energy is **proportional** to execution **time**.

Study 2: Concurrent programming constructs

RQ1. Do alternative **thread management** constructs have different impacts on **energy consumption**?

RQ2. Do alternative **data-sharing primitives** have different impacts on **energy consumption**?

Thread management: `forkIO, forkOn, forkOS`

Data sharing: `MVar, TVar, TMVar`

The Computer Language Benchmarks Game

ROSETTACODE.ORG

9 benchmarks: IO, memory, synchronization bound

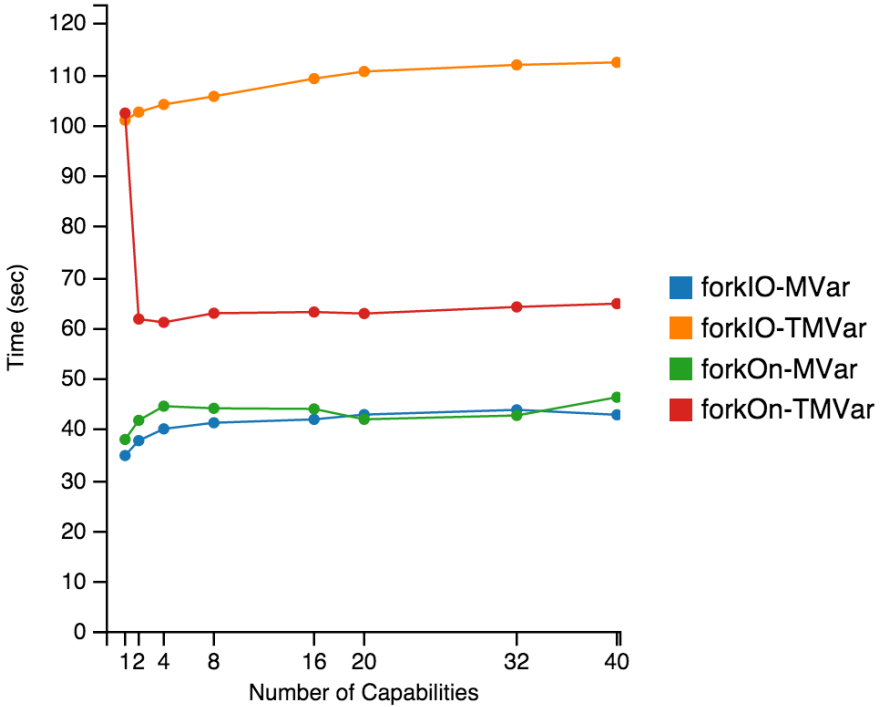
Up to **9 variants** per benchmark

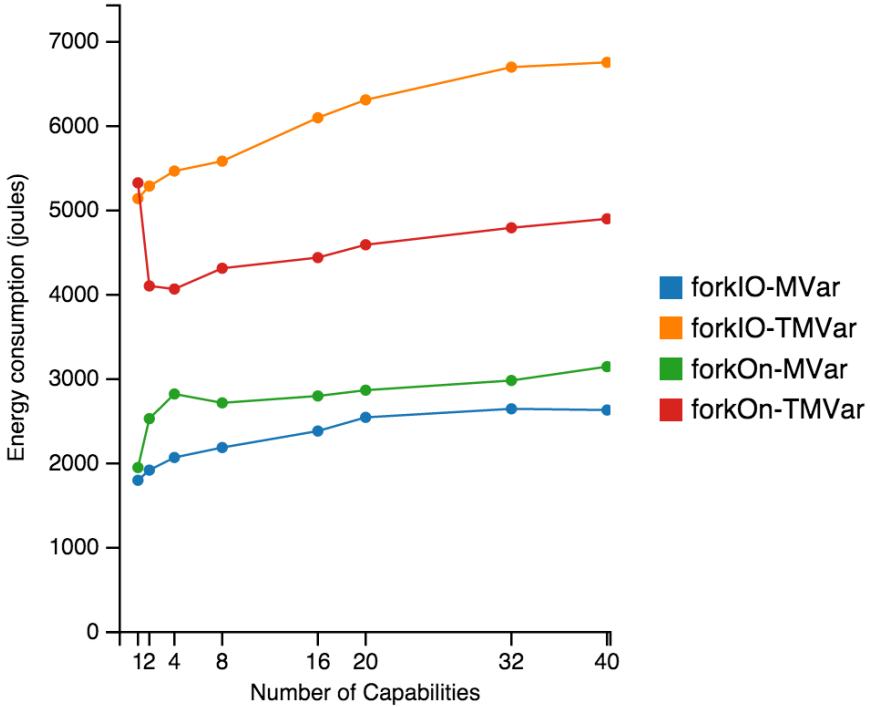
9 configurations for **# of capabilities**

http://green-haskell.github.io

Small changes can produce big savings

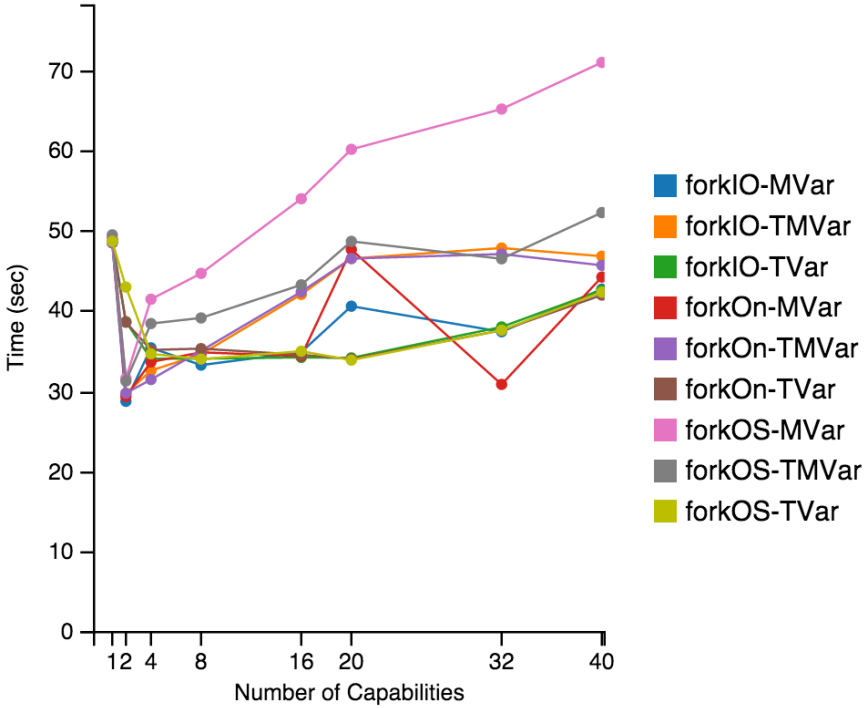
chameneos-redux

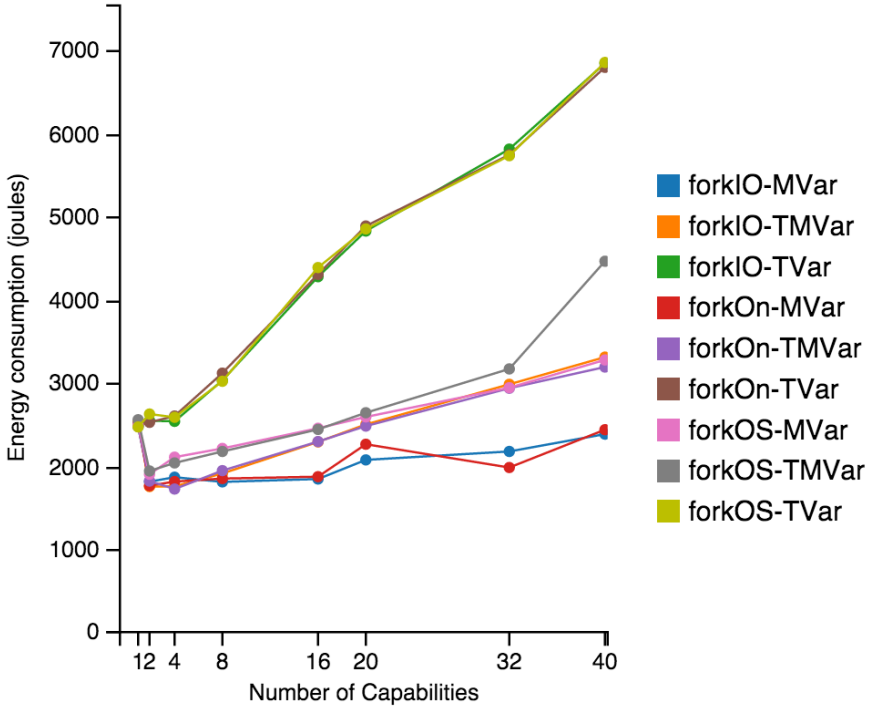




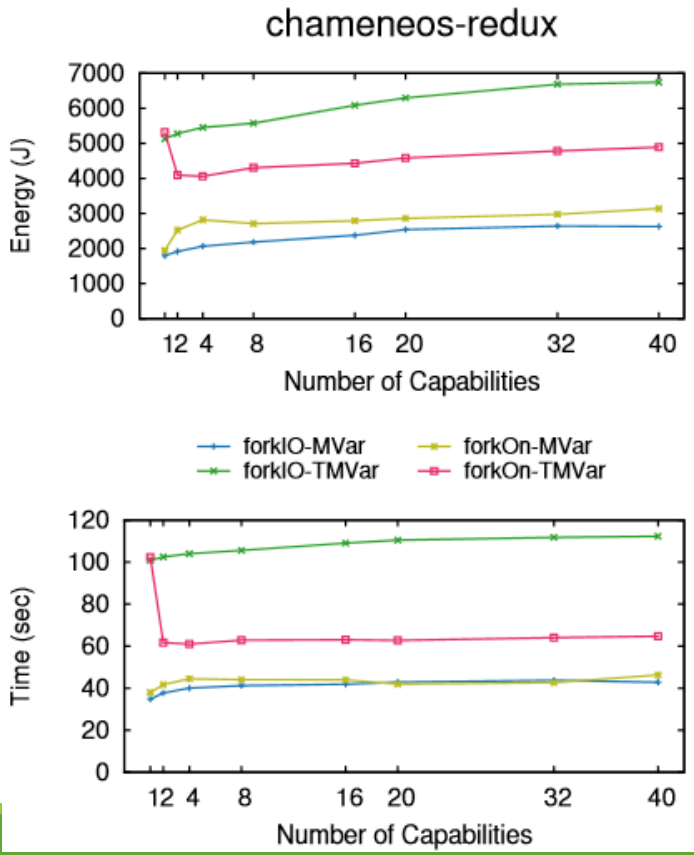
Faster is not always greener

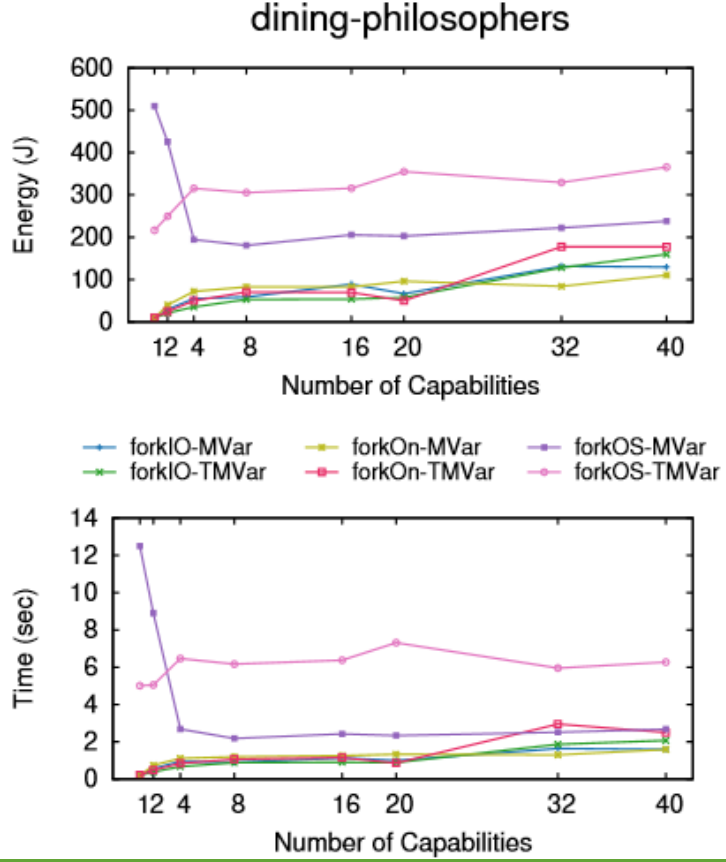
fasta





There is no overall winner





Made two tools energy-aware:

GHC profiler

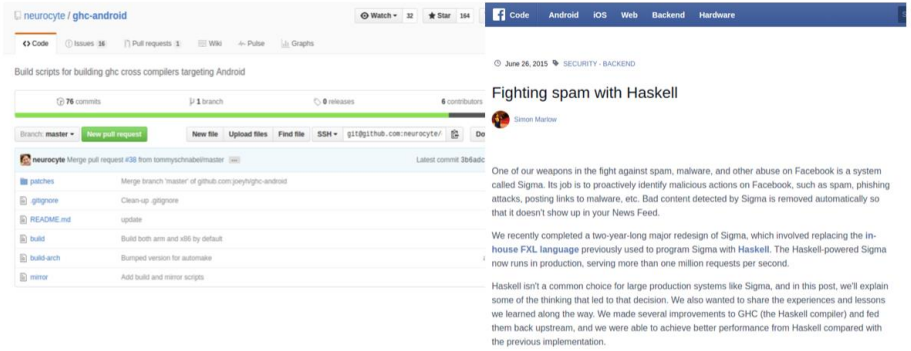
Criterion


Qaanaaq
(Thule)
GREENLAND
Ittoqqortoormiit
(Scoresbysund)
Kulusuk
Ammassalik
Nuuk
(Godthåb)
Narsarsuaq

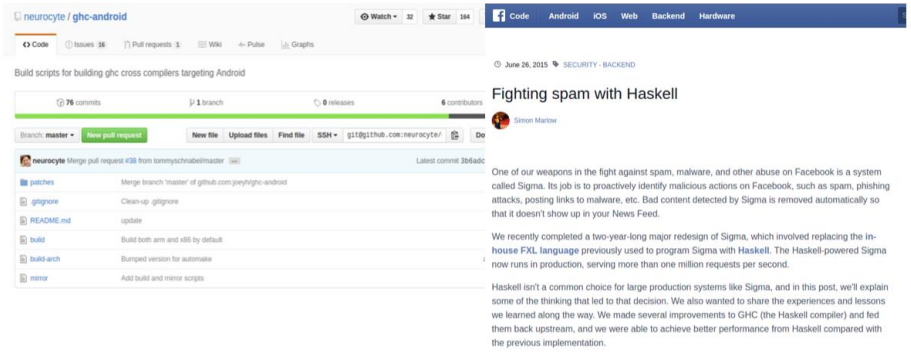
Population: 56483 (2013)

Haskell programmers: 0 (est.)

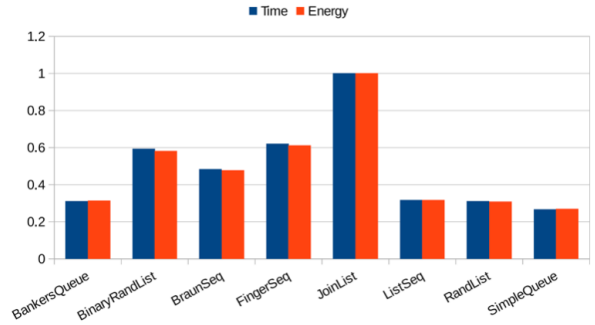
Haskell has feelings too!



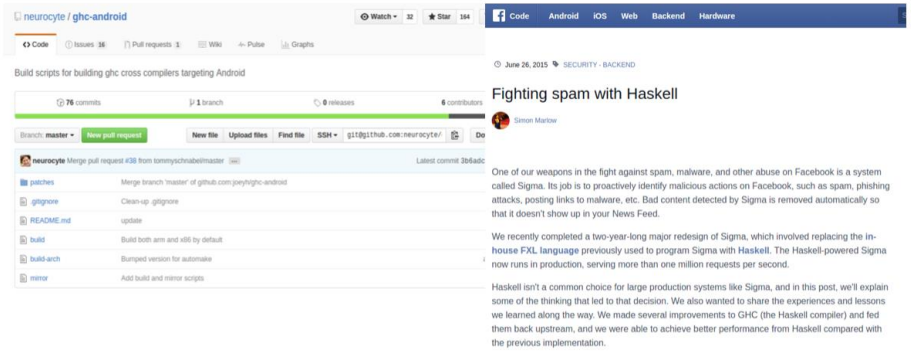
Haskell has feelings too!



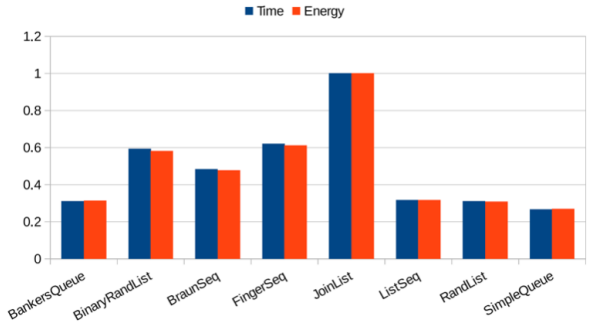
Study 1: Results



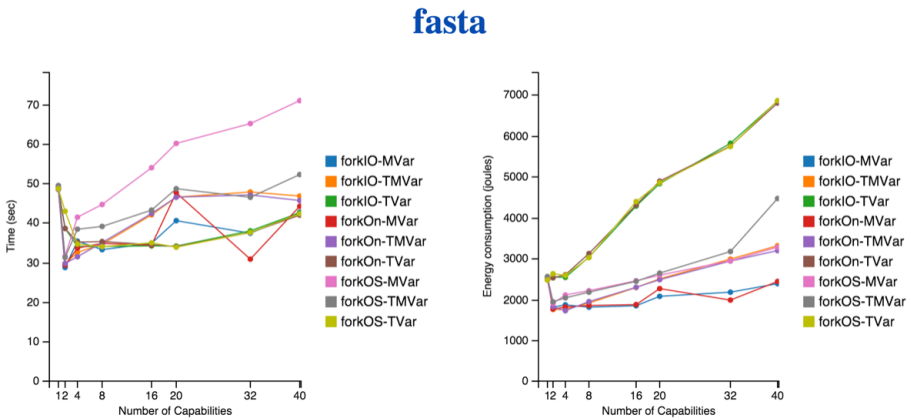
Haskell has feelings too!



Study 1: Results

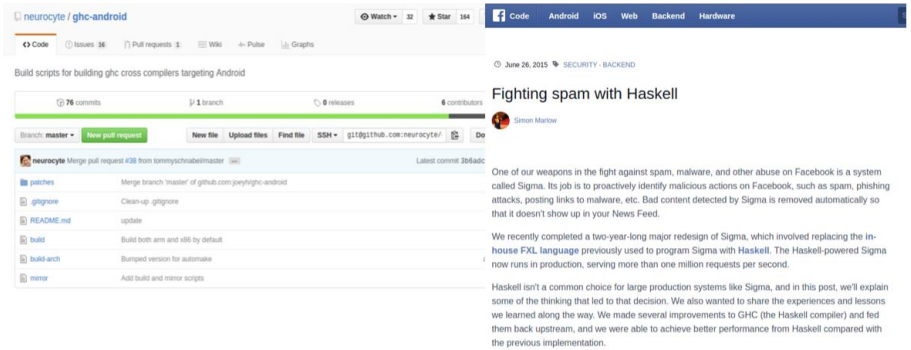


Faster is not always greener

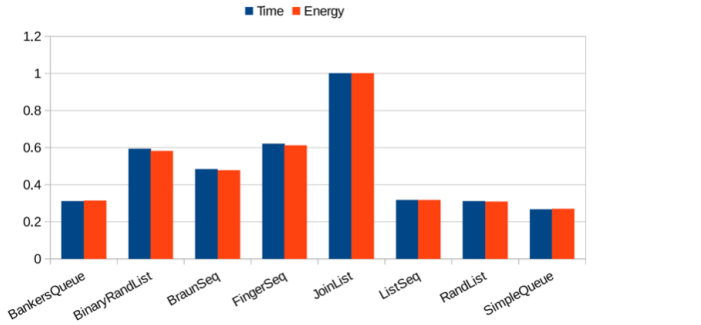


http://green-haskell.github.io

Haskell has feelings too!

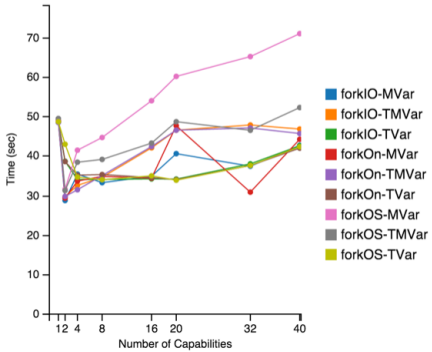


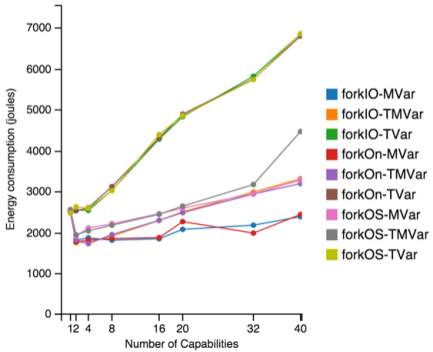
Study 1: Results



Faster is not always greener

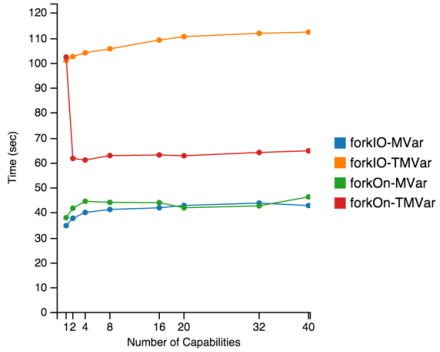
fasta

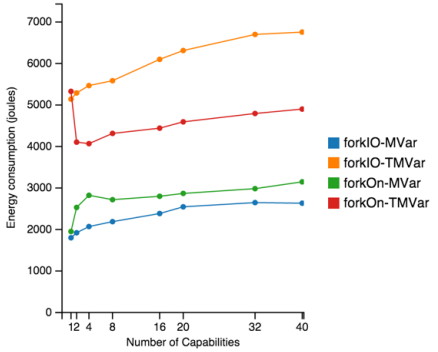




Small changes can produce big savings

chameneos-redux





http://green-haskell.github.io