SCALABLE AND RELIABLE VIDEO TRANSCODING IN HASKELL

Alfredo Di Napoli

Haskell Exchange 2015

ABOUT ME

Full story at: **http://goo.gl/qkKwKm**


```
launchMissile :: IO ()
```

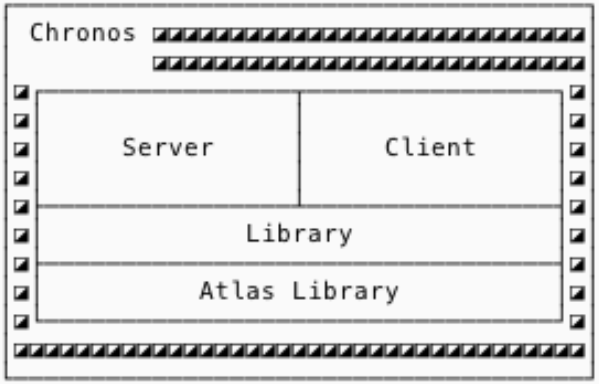
IRIS CONNECT

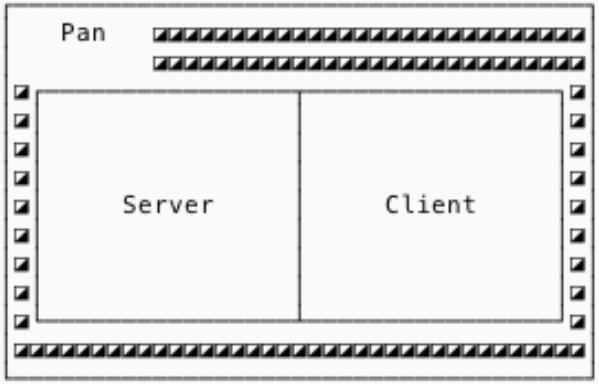
Market leader for CPD solutions

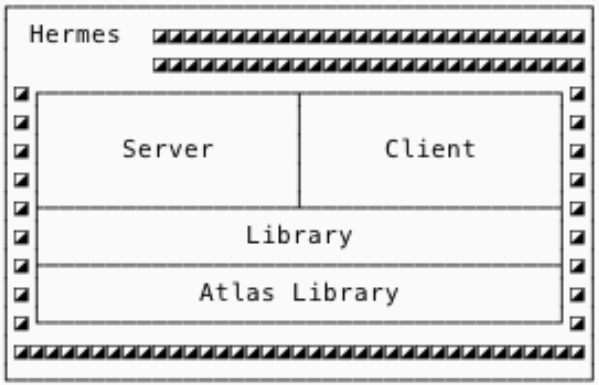
Used by more than 1000 schools across UK, Europe, US & Australia

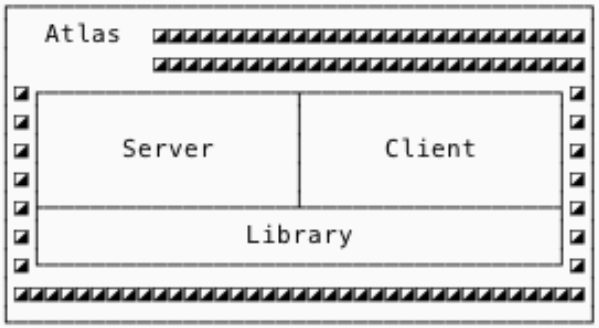
IRIS connect (contd.)

IRIS' GREEK ZOO









HERMES' EVOLUTION

October, 2013

```
hermes [74aeab2] ⚡cloc src scripts
       3 text files.
       3 unique files.
       0 files ignored.

http://cloc.sourceforge.net v 1.60  T=0.24 s (12.3 files/s, 762.9 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Haskell                          1             22              7             80
Bourne Shell                     2             25              2             50
-------------------------------------------------------------------------------
SUM:                             3             47              9            130
-------------------------------------------------------------------------------
```

October, 2015

```
hermes [master] cloc main server src test
     105 text files.
     105 unique files.
       0 files ignored.

http://cloc.sourceforge.net v 1.60  T=0.66 s (159.6 files/s, 26288.3 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Haskell                        105           2332           2252          12712
-------------------------------------------------------------------------------
SUM:                           105           2332           2252          12712
-------------------------------------------------------------------------------
```

Hermes' challenges

Upon taking the lead on Hermes, I was asked for a couple of requirements to be fullfilled, the most important one being that the system needed to be deployed in a cluster, capable of scaling according to demand.

HERMES' CHALLENGES

More specifically, we wanted a system with these desirable properties:

- ~ Scalable
- ~ Fault tolerant
- ~ Distributed

"Out of the Tar Pit" docet

> The classical ways to approach the difficulty of state include OOP programming which tightly couples state together with related behaviour, and functional programming which — in its pure form — eschews state and side-effects all together. [..] We argue that it is possible to take useful ideas from both and that this approach offers significant potential for simplifying the construction of large-scale software systems.

In the same fashion, we have 2 different worlds colliding:

- ~ We need to transcode videos, which is a very stateful operation
- ~ As good Haskell programmers, we want to have components in our system to be as stateless as possible.

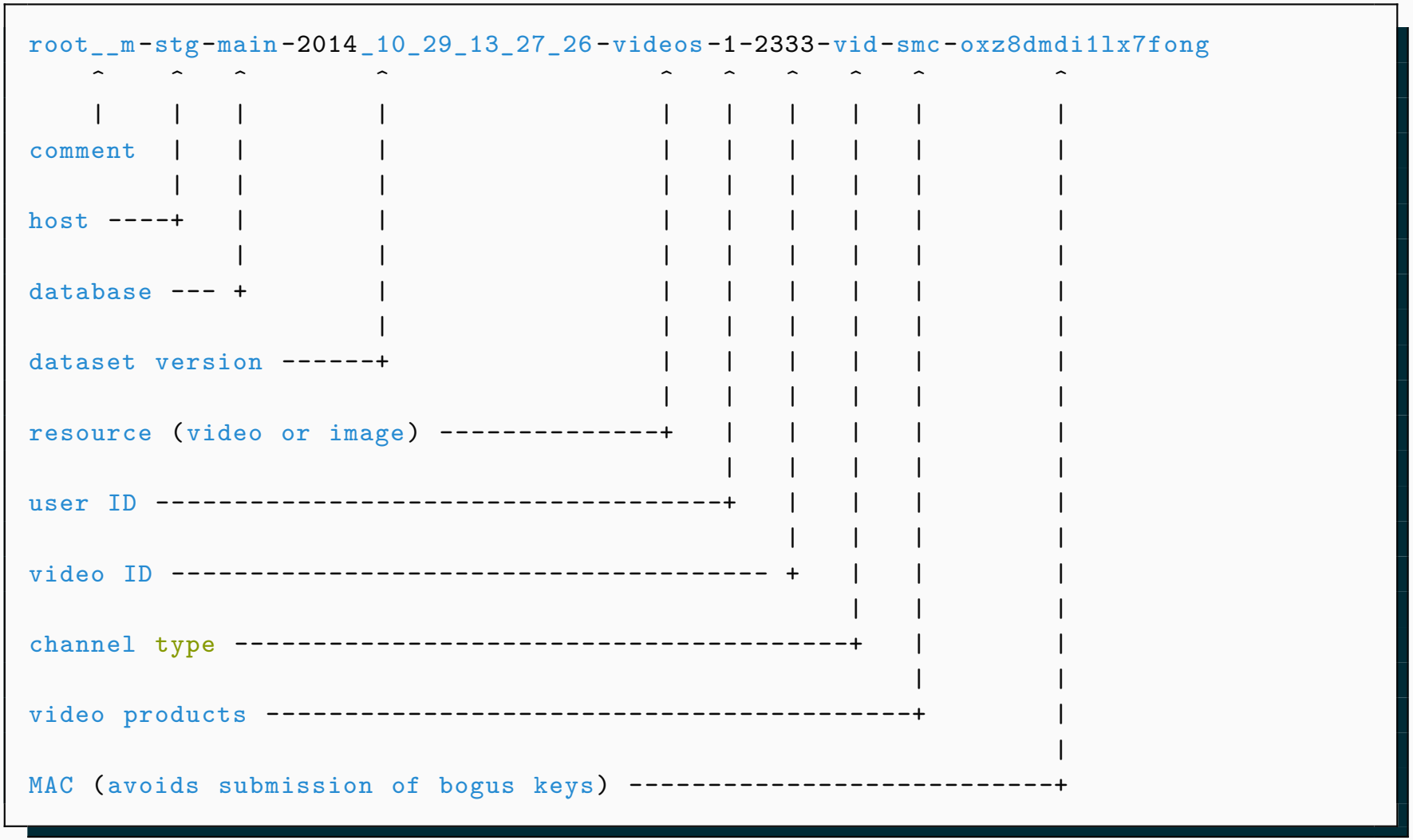
A SHARED NOTHING ARCHITECTURE (SN)

A shared nothing architecture (SN) is a distributed computing architecture in which each node is independent and self-sufficient, and there is no single point of contention across the system.

"All problems in computer science can be solved by another level of indirection." - Butler Lampson

1. RabbitMQ was just the right tool for the job at hand:
   - ~ Easy to setup
   - ~ Can be configured to operate in a federation of nodes
   - ~ Extremely reliable
   - ~ Good Haskell bindings for it (*AMQP*)
2. A question genuinely arise: it seems extremely costly to shuffle video as binary blobs over the queues. Can we avoid that?

ABSTRACTION IS THE (MEDIA KEY)



To be fair, the media key abstraction was already present in Atlas when I choose RabbitMQ, but it was the perfect fit for it!

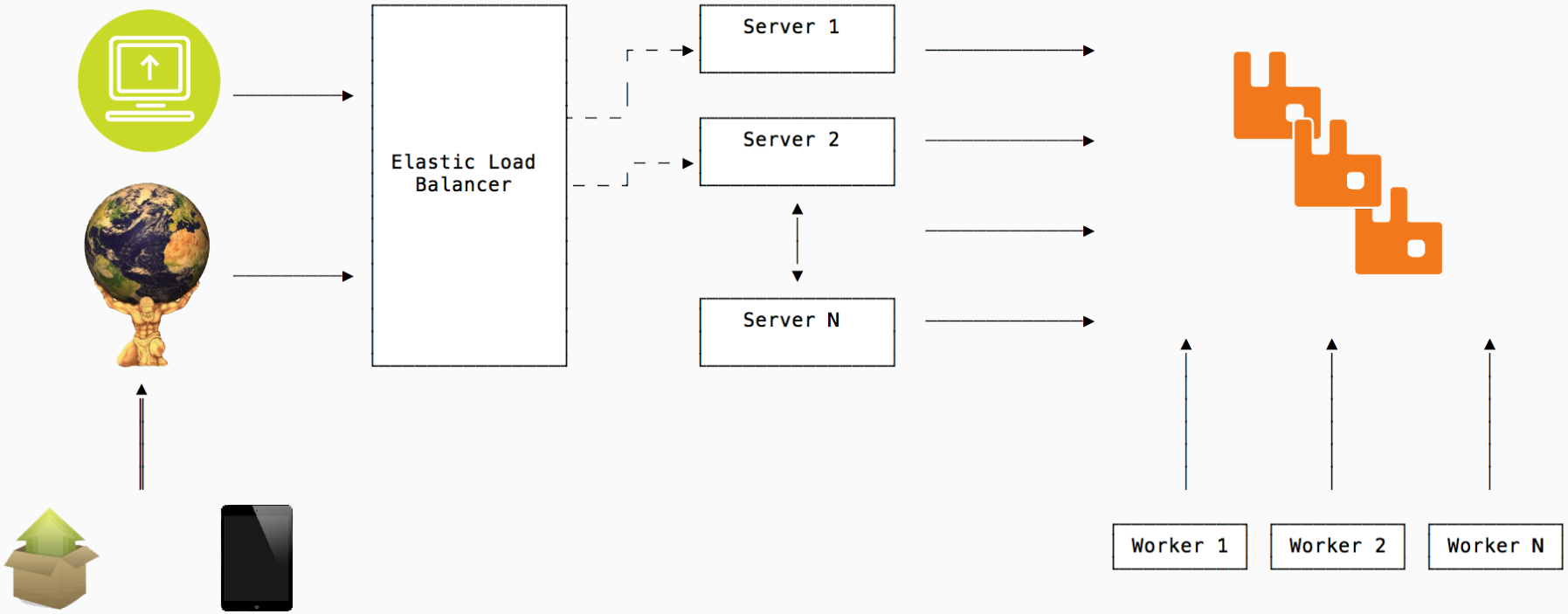
WHAT ABOUT SCALABILITY?

Fine, but RabbitMQ doesn't give you scalability...

1. We stood on the shoulder of giants - namely AWS' Auto Scaling Groups
2. Our very first native scaling algorith looked like:
   - ~ Scaling up: Based on CPU% over time
   - ~ Scaling down: Based on CPU% over time

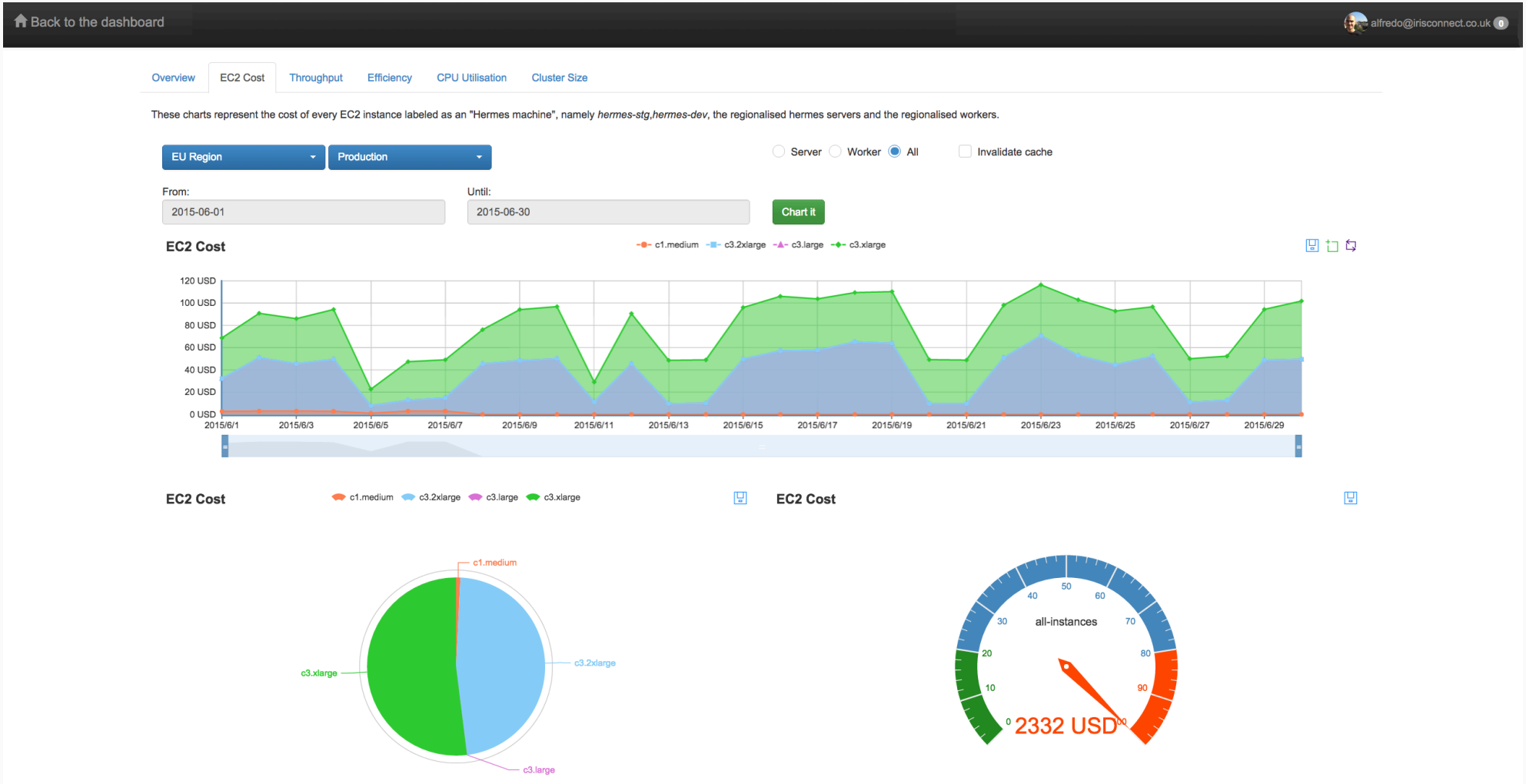
It kept us going for a while...

THE ARCHITECTURE



REVIEWING THE SCALING EXPERIENCE

1. Scaling up was too conservative and slow
   - ~ It could take up to 15 mins to spawn a new worker
2. Scaling down suffered similar problems
3. The result was unoptimal customer experience (due to the slow turnaround time) and unoptimal for us (due to the additional costs incurring from poor scaling down)


Back to the dashboard
alfredo@irisconnect.co.uk
Overview
EC2 Cost
Throughput
Efficiency
CPU Utilisation
Cluster Size
These charts represent the cost of every EC2 instance labeled as an "Hermes machine", namely hermes-stg,hermes-dev, the regionalised hermes servers and the regionalised workers.
EU Region
Production
Server
Worker
All
Invalidate cache
From:
2015-06-01
Until:
2015-06-30
Chart it
EC2 Cost
c1.medium
c3.2xlarge
c3.large
c3.xlarge
120 USD
100 USD
80 USD
60 USD
40 USD
20 USD
0 USD
2015/6/1
2015/6/3
2015/6/5
2015/6/7
2015/6/9
2015/6/11
2015/6/13
2015/6/15
2015/6/17
2015/6/19
2015/6/21
2015/6/23
2015/6/25
2015/6/27
2015/6/29
EC2 Cost
c1.medium
c3.2xlarge
c3.large
c3.xlarge
EC2 Cost
c1.medium
c3.xlarge
c3.2xlarge
c3.large
all-instances
2332 USD

What's the elephant in the room?

Why not use Cloud Haskell?

WHY NOT CLOUD HASKELL

1. CH encourages Erlang-style (i.e. actor based) communication, so nodes should know each other

   We do not want that!

2. Peer discovery would have been tricky in a dynamic environment where new machines born and die frequently
3. It wasn't mature enough in 2013, if not for a handful of companies using it

Thank you!

Questions?

My road to Haskell

http://www.alfredodinapoli.com/posts/2014-04-27-my-road-to-haskell.html

Don Stewart - Haskell in the large

http://code.haskell.org/~dons/talks/dons-google-2015-01-27.pdf