

Scaling Research in the Cloud

Research Computing @ AWS

AWS Worldwide Research Computing



Who's Who

Ralph



Snr Solution Architect

- Leads solution designing with AWS Pub Sec customers all over DE, AT & CH
- Software Engineer
- Based in Munich
- Owns a cat

Scott



HPC Specialist

- Recovering Professor
- Aircraft Designer
- Based in London

Boof



Research Computing Manager

- Recovering Physicist & Super Computer Guy
- Based in London
- Owns a dog

< 1 MIN >

Ralph: Alces Flight Cluster **launch**

**IT'S ABOUT
SCIENCE,
NOT
SERVERS.**

aws.amazon.com/rcp

[#AWSresearchcloud](https://twitter.com/AWSresearchcloud)



DATASETS, TOOLS & TECHNIQUES

aws.amazon.com/rcp

#AWSresearchcloud



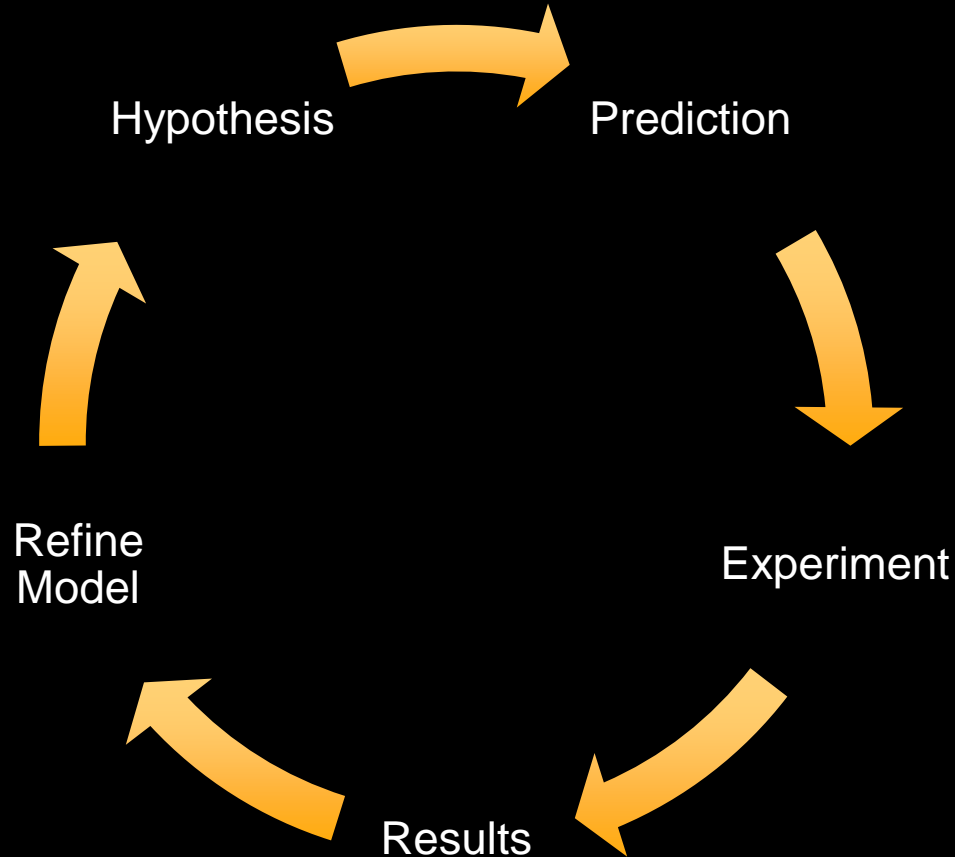
Experimentation (& Failure) is essential

“Invention requires two things: the ability to try a **lot of experiments**, and not having to live with the collateral damage of **failed experiments**.”

*Andy Jassy**

* The guy who pays my salary.

The Scientific Method

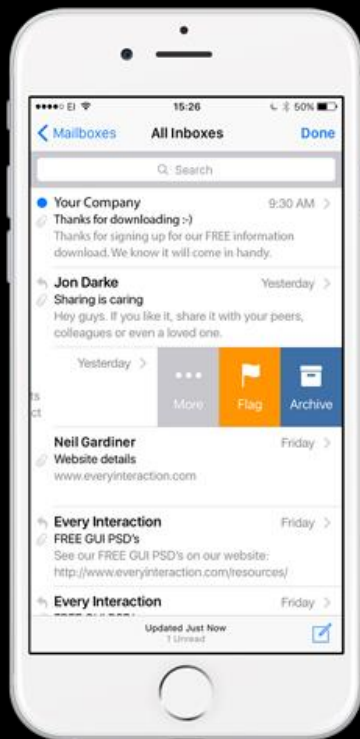


```
~ [10] $ aws s3 ls s3://publicboof/onHoldMusic/
2017-03-22 23:54:19 David Hasselhoff - All the Right Moves
2016-10-25 16:17:39 David Hasselhoff - Amazing Grace
2016-10-07 00:39:49 David Hasselhoff - America
2015-11-27 04:24:53 David Hasselhoff - Any Kind of Love at All
2015-11-27 04:25:55 David Hasselhoff - Blue Bayou
2017-05-04 23:26:04 David Hasselhoff - California Dreaming
2017-11-15 13:12:12 David Hasselhoff - California Girls
2016-12-03 00:53:58 David Hasselhoff - City Of New Orleans
2017-04-25 19:16:41 David Hasselhoff - Country Roads
2016-05-16 22:53:53 David Hasselhoff - Crazy on a Saturday Night
2016-05-16 22:53:51 David Hasselhoff - Dark Side Of My Heart
2016-06-25 01:56:13 David Hasselhoff - Days Of Our Love
2017-03-08 20:21:38 David Hasselhoff - Do You Believe In Love
2017-07-13 02:53:34 David Hasselhoff - Do You Love Me?
2017-03-10 03:16:33 David Hasselhoff - Fallin' In Love
~ [10] $ aws s3 cp s3://publicboof/onHoldMusic/
```




We don't do this

```
$ telnet example.org 25
S: 220 example.org ESMTP Sendmail 8.13.1/8.13.1; Wed, 30 Aug 2006
07:36:42 -0400
C: HELO mailout1.phrednet.com
S: 250 example.org Hello ip068.subnet71.gci-net.com [216.183.71.68],
pleased to meet you
C: MAIL FROM:<xxxx@example.com>
S: 250 2.1.0 <xxxx@example.com>... Sender ok
C: RCPT TO:<yyyy@example.com>
S: 250 2.1.5 <yyyy@example.com>... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
From: Dave\r\nTo: Test Recipient\r\nSubject: SPAM SPAM SPAM\r\n\r\nThis
is message 1 from our test script.\r\n.\r\n
S: 250 2.0.0 k7TKIBYb024731 Message accepted for delivery
C: QUIT
S: 221 2.0.0 example.org closing connection
Connection closed by foreign host.
```



It can hardly be a coincidence that no language on Earth has ever produced the expression "**As pretty as an airport.**"

Douglas Adams

No one ever said a 'bsub' script was pretty either.

```
#!/bin/bash
#SBATCH --job-name=gpuMemTest
#SBATCH --output=gpuMemTest_%j.out
#SBATCH --error=gpuMemTest_%j.err
#SBATCH --ntasks=2
#SBATCH --cpus-per-task=1
#SBATCH --distribution=cyclic:cyclic
#SBATCH --time=12:00:00
#SBATCH --mem-per-cpu=2000
##SBATCH --mail-type=END,FAIL
##SBATCH --mail-user=email@ufl.edu
#SBATCH --partition=gpu
#SBATCH --gres=gpu:tesla:2
date;hostname;pwd

module load cuda/9.1.85

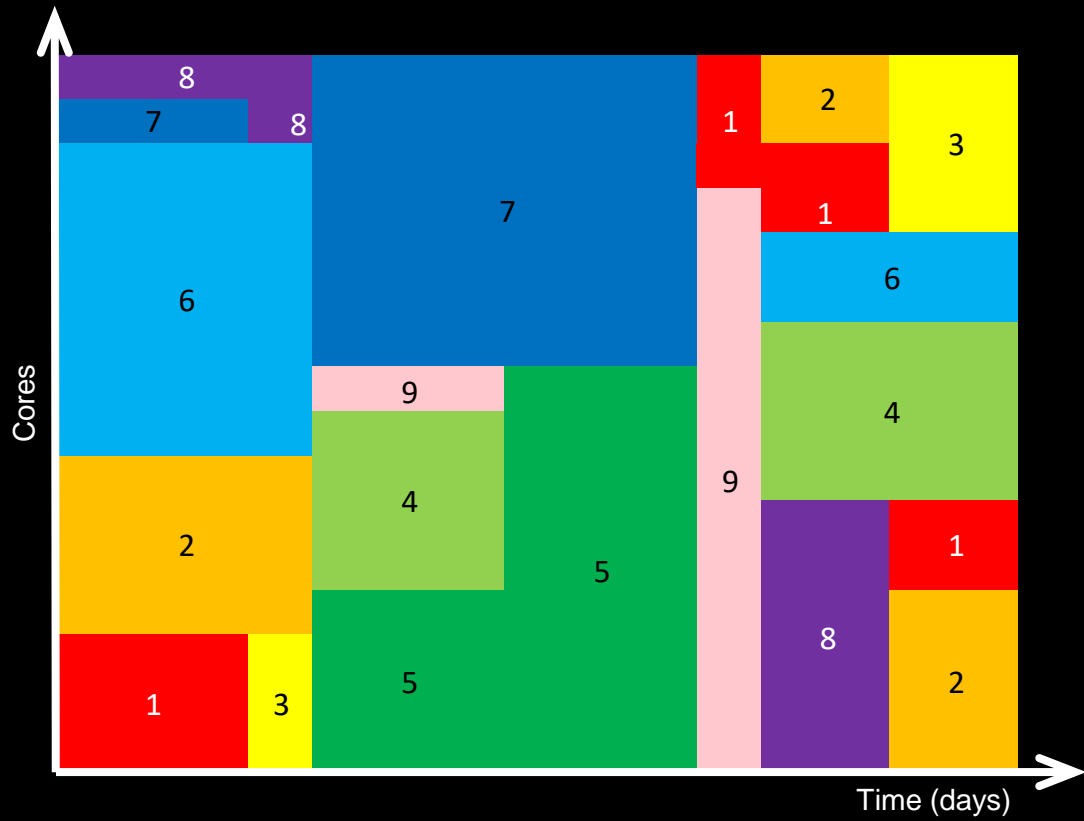
cudaMemTest=/ufrc/ufhpc/chasman/Cuda/cudaMemTest/cuda_memtest

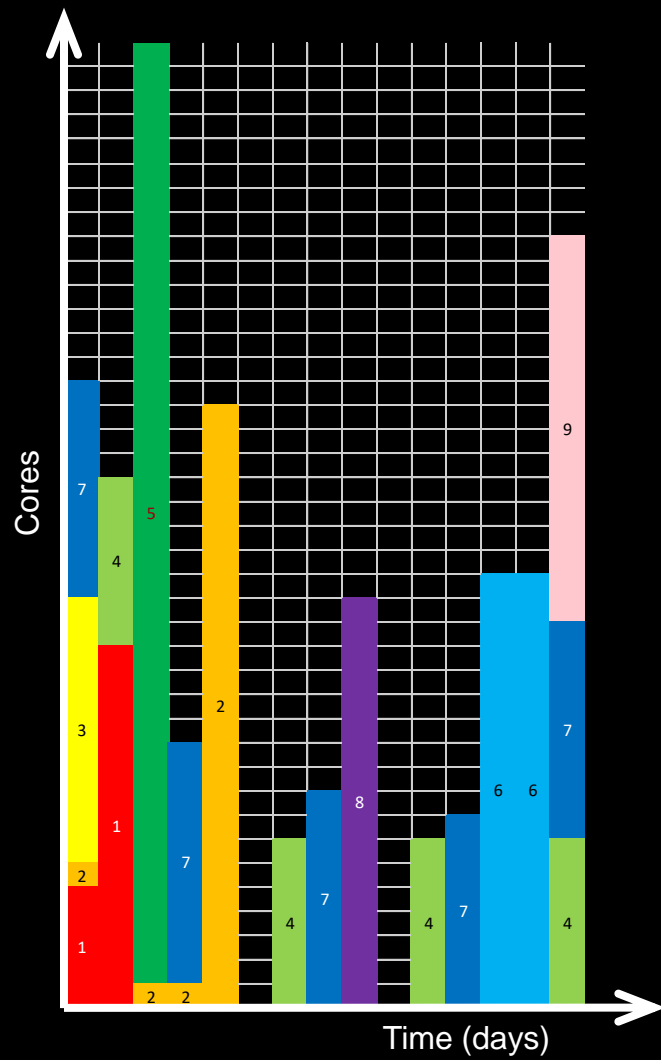
cudaDevs=$(echo $CUDA_VISIBLE_DEVICES | sed -e 's/,/ /g')

for cudaDev in $cudaDevs
do
    echo cudaDev = $cudaDev
    #srun --gres=gpu:tesla:1 -n 1 --exclusive ./gpuMemTest.sh >
gpuMemTest.out.$cudaDev 2>&1 &
    $cudaMemTest --num_passes 1 --device $cudaDev > gpuMemTest.out.$cudaDev 2>&1 &
done
```



Are we solving the right problem?



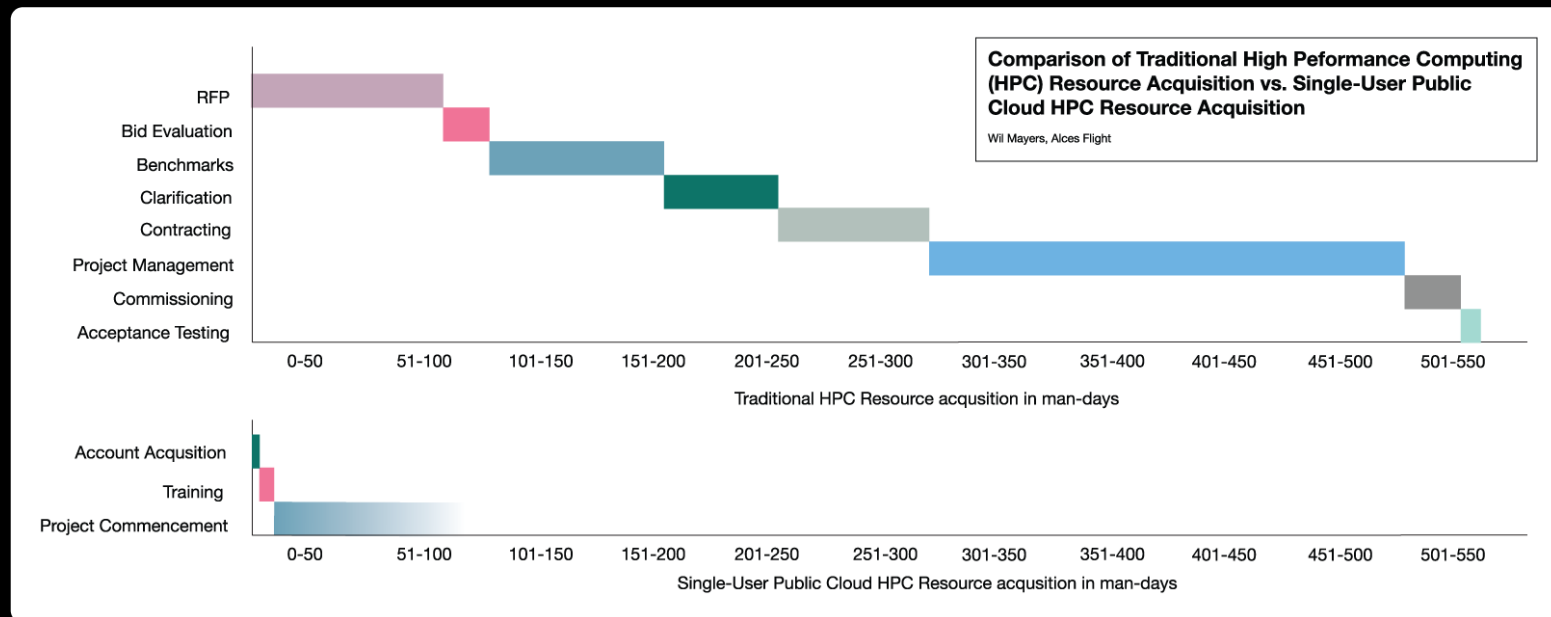


< 1 MIN >

Ralph: Cluster **job submit**

Are we iterating fast enough?

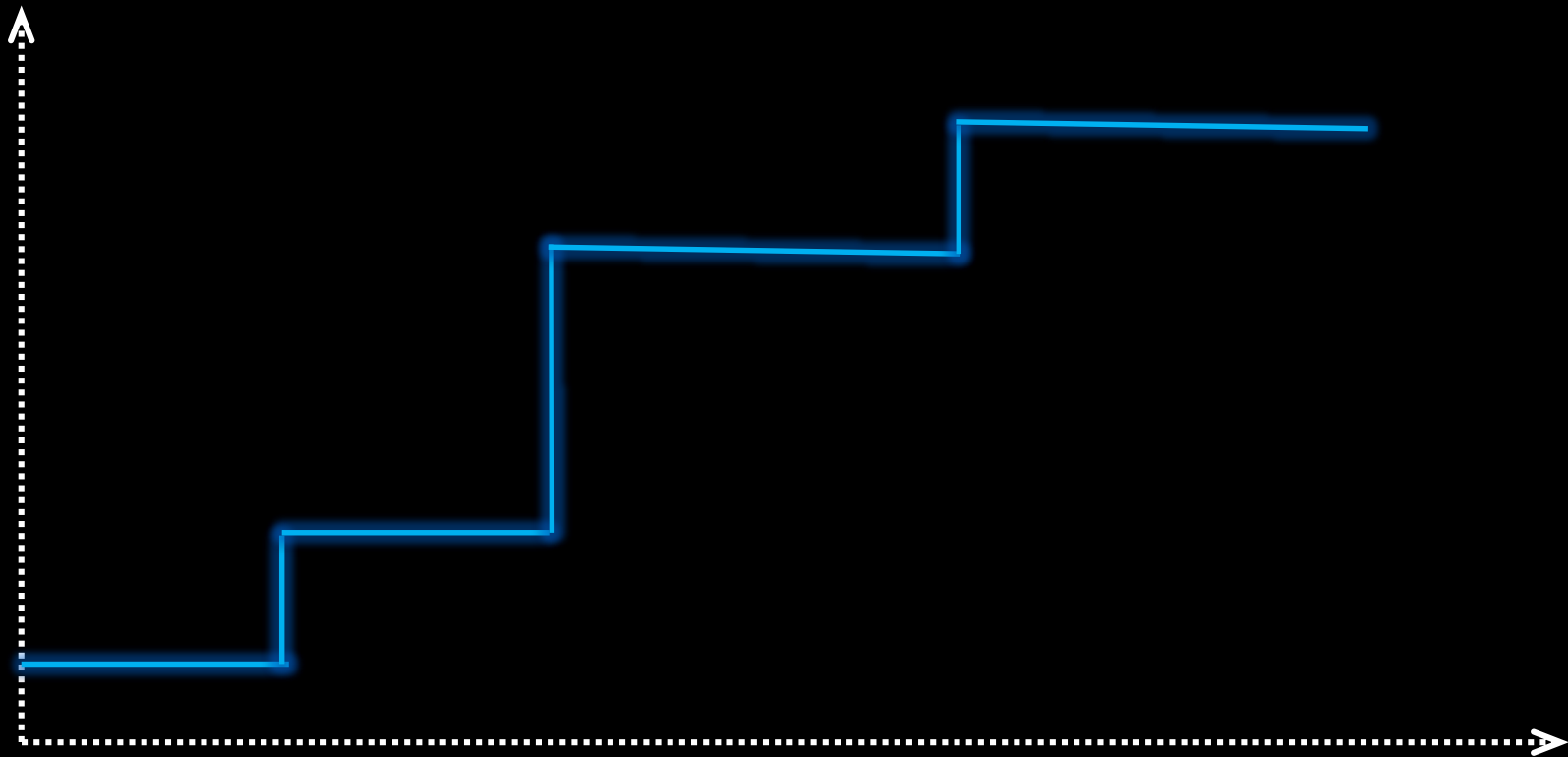
RFP != INNOVATION



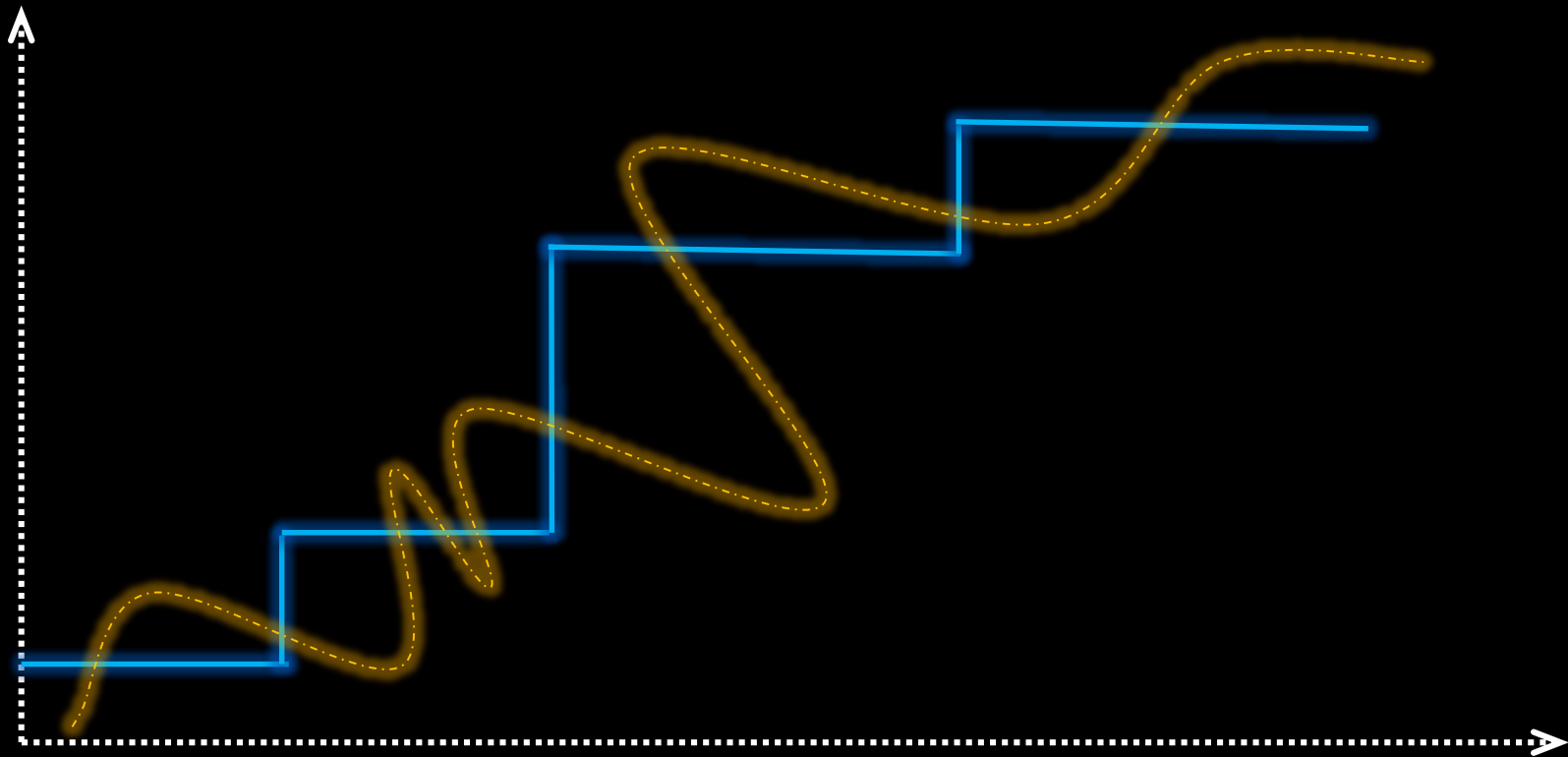
Are we solving problems for actual scientists?

Or are we totally doing an awesome job of getting a great score on a benchmark which we think is meaningful?

Hardware capability moves in leaps and bounds

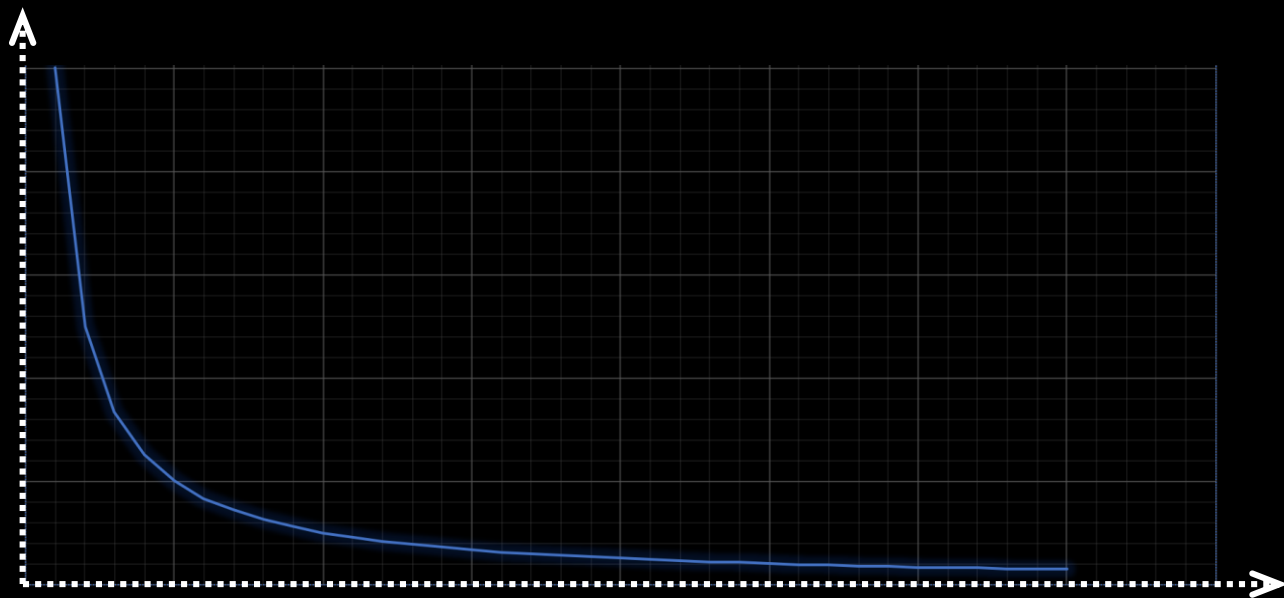


Humans don't

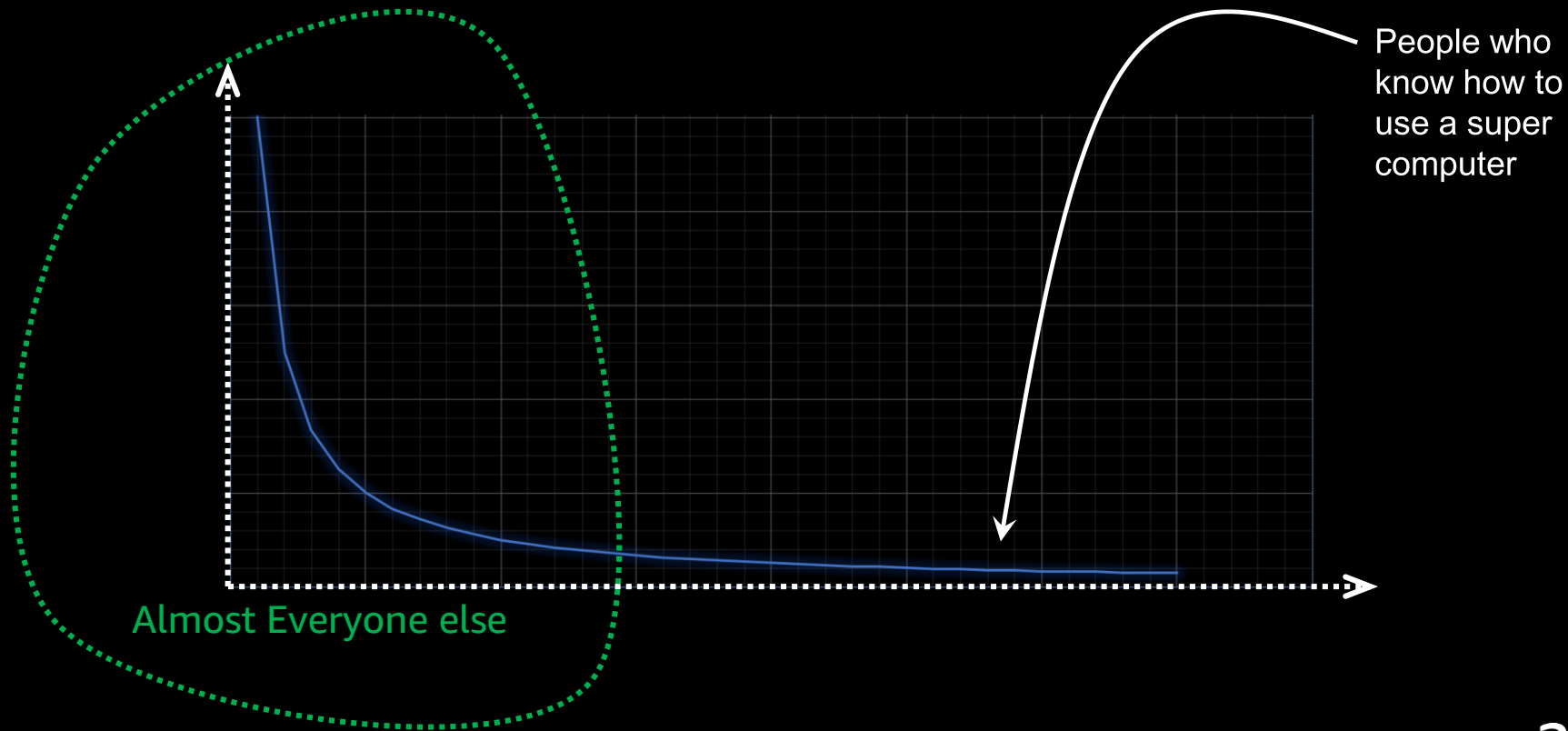


Who are we solving for?

Our community



Our community



PRABHU ET AL (2009)

"Despite enormous wait times, many scientists run their programs only on desktops"

"About a third of researchers did not use any form of parallelism in their research at all"

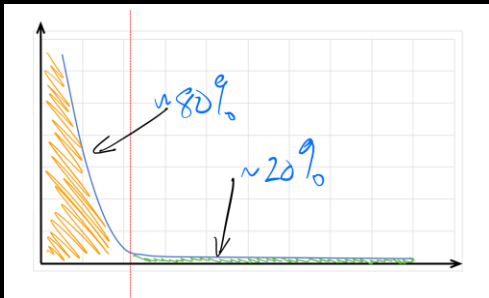
"Currently, many researchers fit their scientific models to only a subset of available parameters for faster program runs."

HANNAY ET AL (2009)

- Online survey of 1972 international researchers
- ~80% never use a supercomputer

Getting [IT] out of the way

Technology needs to be in the **service** of the science, not it's master.



Most researchers have needs that fall short of a National Supercomputer, but quickly become bigger than their laptop.

But **traditionally** that next step up meant getting consumed in server CPU specifications and **learning job submission syntax** or guessing which hard drive geometries offers enough IOPS.



Ian Hawke

@IanHawke

Follow



Here we go with [@gvwilson](#) : the gulf between the computing scientific "elite" and those emailing spreadsheets is growing, and that's bad.

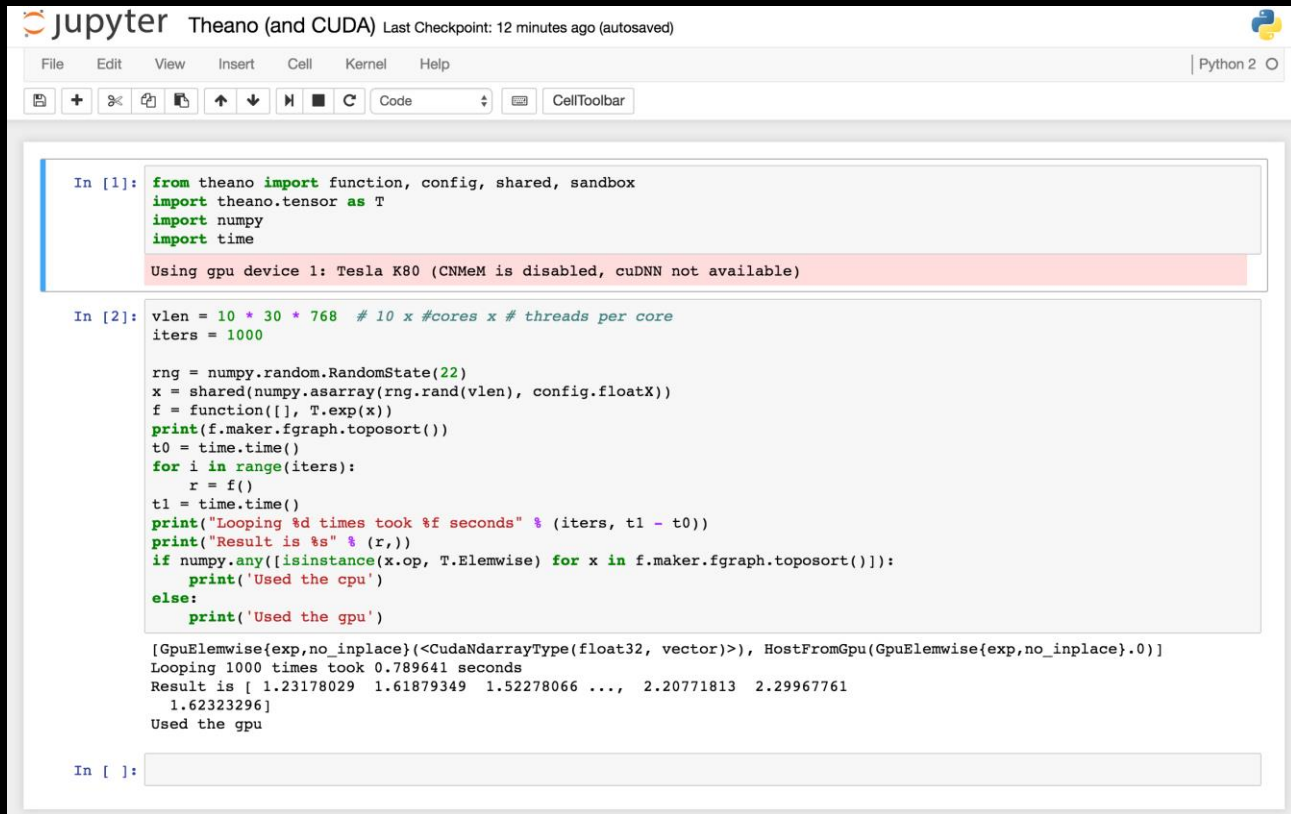
2:06 PM - 10 Nov 2015

Scaling up should be about quickly adding extra capacity, performance, memory or storage quickly, and being able to **scale down** just as quick when it's time to put the work on ice for the weekend.

Sharing the latest methods with collaborators should mean sending them a link to a whole machine full of working software. And receiving one back, with **something new inside**.

Scale

Immediate scale factor



The screenshot shows a Jupyter Notebook titled "Theano (and CUDA)" with a last checkpoint of 12 minutes ago. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for saving, adding cells, and running code. The code is written in Python 2. The first cell (In [1]) imports Theano, NumPy, and time, and displays a message about using GPU device 1 (Tesla K80). The second cell (In [2]) defines a function to calculate the exponential of a vector, loops 1000 times, and prints the result and time taken. The output shows that the GPU was used and the calculation took 0.789641 seconds.

```
In [1]: from theano import function, config, shared, sandbox
import theano.tensor as T
import numpy
import time

Using gpu device 1: Tesla K80 (CNMeM is disabled, cuDNN not available)

In [2]: vlen = 10 * 30 * 768 # 10 x #cores x # threads per core
iters = 1000

rng = numpy.random.RandomState(22)
x = shared(numpy.asarray(rng.rand(vlen), config.floatX))
f = function([], T.exp(x))
print(f.maker.fgraph.toposort())
t0 = time.time()
for i in range(iters):
    r = f()
t1 = time.time()
print("Looping %d times took %f seconds" % (iters, t1 - t0))
print("Result is %s" % (r,))
if numpy.any([isinstance(x.op, T.Elemwise) for x in f.maker.fgraph.toposort()]):
    print('Used the cpu')
else:
    print('Used the gpu')

[GpuElemwise{exp,no_inplace}<CudaNdarrayType(float32, vector)>, HostFromGpu(GpuElemwise{exp,no_inplace}.0)]
Looping 1000 times took 0.789641 seconds
Result is [ 1.23178029  1.61879349  1.52278066 ...,  2.20771813  2.29967761
 1.62323296]
Used the gpu

In [ ]:
```

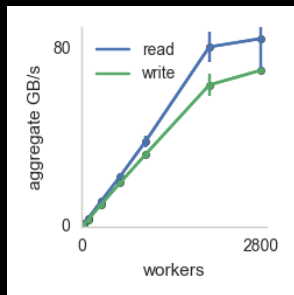
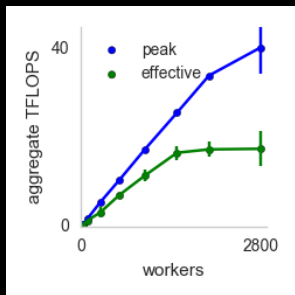
Scale:

- From **laptop** to **server**
- From **server** to **cluster**
- From **CPU** to **GPU**

... in minutes.

PyWren lets you run your existing python code at massive scale via AWS Lambda

```
def my_function(b):  
    x = np.random.normal(0, b, 1024)  
    A = np.random.normal(0, b, (1024, 1024))  
    return np.dot(A, x)  
  
pwex = pywren.default_executor()  
res = pwex.map(my_function, np.linspace(0.1, 100, 1000))
```



Occupy the Cloud: Distributed Computing for the 99%

Eric Jonas, Qifan Pu, Shivaram Venkataraman, Ion Stoica, Benjamin Recht

University of California, Berkeley

Submission Type: Vision

Abstract

Distributed computing remains inaccessible to a large number of users, in spite of many open source platforms and extensive commercial offerings. While distributed computation frameworks have moved beyond a simple map-reduce model, many users are still left to struggle with complex cluster management and configuration tools, even for running simple embarrassingly parallel jobs. We argue that stateless functions represent a viable platform for these users, eliminating cluster management overhead, fulfilling the promise of elasticity. Furthermore, using our prototype implementation, PyWren, we show that this model is general enough to implement a number of distributed computing models, such as BSP, efficiently. Extrapolating from recent trends in network bandwidth and the advent of disaggregated storage, we suggest that stateless functions are a natural fit for data processing in future computing environments.

framework. Yet even at UC Berkeley, we have found via informal surveys that the majority of machine learning graduate students have never written a cluster computing job due to complexity of setting up cloud platforms.

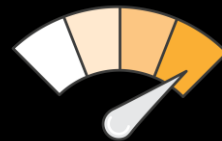
In this paper we argue that a *serverless* execution model with *stateless* functions can enable radically-simpler, fundamentally elastic, and more user-friendly distributed data processing systems. In this model, we have one simple primitive: users submit functions that are executed in a remote container; the functions are stateless as all the state for the function, including input, output is accessed from shared remote storage. Surprisingly, we find that the performance degradation from using such an approach is negligible for many workloads and thus, our simple primitive is in fact general enough to implement a number of higher-level data processing abstractions, including MapReduce and parameter servers.

Recently cloud providers (e.g., AWS Lambda, Google Cloud Functions) and open source projects (e.g., OpenLambda [16], OpenWhisk [30]) have developed infras-

< 5 MIN >

Ralph: DEMO of PyWren

Scaling Research in a Hybrid Environment



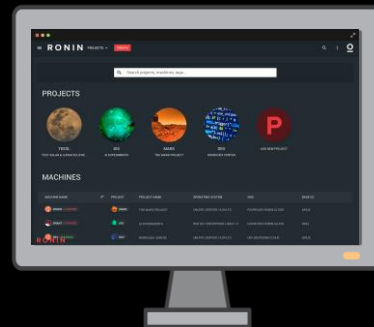
LAPTOP



Most research starts here.

Unfortunately, most research remains trapped here due to the complexity of accessing scalable computing resources.*

CLOUD



Immediately scale workloads to larger servers, with more cores, faster I/O, more memory. Scale up, or down, come and go easily, with tight budget controls.



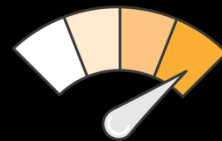
Create your own software stacks that can be re-used and shared instantly.

Curate a **local catalog** of pre-packaged offerings ready to go (eg Matlab running campus license)



* SOURCE: HANNAY ET AL (2009)

Scaling Research in a Hybrid Environment



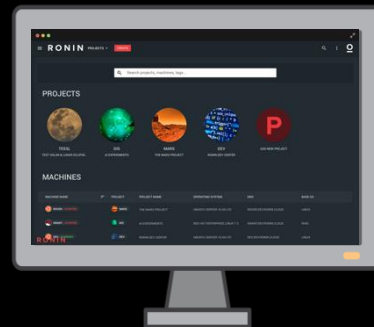
LAPTOP



Most research starts here.

Unfortunately, most research remains trapped here due to the complexity of accessing scalable computing resources.*

RONIN



Immediately scale workloads to larger servers, with more cores, faster I/O, more memory. Scale up, or down, come and go easily, with tight budget controls.



Create your own software stacks that can be re-used and shared instantly.

Curate a **local catalog** of pre-packaged offerings ready to go (eg Matlab running campus license)



* SOURCE: HANNAY ET AL (2009)

AWS services

Find a service by name or feature (for example, EC2, S3 or VM, storage).

Recently visited services

- IAM
- EC2

All services

- Compute
 - EC2
 - Lightsail
 - Elastic Container Service
 - Lambda
 - Batch
 - Elastic Beanstalk
- Storage
 - S3
 - Glacier
 - Storage Gateway
- Database
 - RDS
 - DynamoDB
 - ElastiCache
 - Neptune
 - Amazon Redshift
- Migration
 - AWS Migration Hub
 - Application Discovery Service
 - Database Migration Service
 - Server Migration Service
 - Snowball
- Networking & Content Delivery
 - VPC
 - CloudFront
- Analytics
- Management Tools
 - AWS Auto Scaling
 - CloudFormation
 - CloudTrail
 - Config
 - OpsWorks
 - Service Catalog
 - Systems Manager
 - Trusted Advisor
 - Managed Services
- Media Services
 - Elastic Transcoder
 - Kinesis Video Streams
 - MediaConvert
 - MediaLive
 - MediaPackage
 - MediaStore
 - MediaTailor
- Machine Learning
 - Amazon SageMaker
 - Amazon Comprehend
 - AWS DeepLens
 - Amazon Lex
 - Machine Learning
 - Amazon Polly
 - Rekognition
 - Amazon Transcribe
 - Amazon Translate
- Business Productivity
 - Alexa for Business
 - Amazon Chime
 - WorkDocs
 - WorkMail
- Desktop & App Streaming
 - WorkSpaces
 - AppStream 2.0
- AR & VR
 - Amazon Sumerian
- Application Integration
 - Step Functions
 - Amazon MQ
 - Simple Notification Service
 - Simple Queue Service
 - SWF
- Customer Engagement
 - Amazon Connect
 - Pinpoint
 - Simple Email Service

Helpful tips

Manage your costs
Monitor your AWS costs, usage, and reservations using AWS Budgets. [Start now](#)

Create an organization
Use AWS Organizations for policy-based management of multiple AWS accounts. [Start now](#)

Explore AWS

Amazon Relational Database Service (RDS)
RDS manages and scales your database for you. RDS supports Aurora, MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server. [Learn more](#)

US East (N. Virginia)
US East (Ohio)
US West (N. California)
US West (Oregon)
Asia Pacific (Mumbai)
Asia Pacific (Osaka-Local)
Asia Pacific (Seoul)
Asia Pacific (Singapore)
Asia Pacific (Sydney)
Asia Pacific (Tokyo)
Canada (Central)
EU (Frankfurt)
EU (Ireland)
EU (London)
EU (Paris)
South America (São Paulo)

Real-Time Analytics with Amazon Kinesis

Launch Instance **Connect** **Actions**

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key
	i-0908fc43909bba2d	m4.10xlarge	eu-west-1a	shutting-down	None		ec2-34-245-199-85.eu-west-1.compute.amazonaws.com	34.245.199.85	-	boof
www.boof.io	i-0110d07c07d17f68d	t2.small	eu-west-1b	running	2/2 checks ...	None	ec2-52-30-80-243.eu-west-1.compute.amazonaws.com	52.30.80.243	-	boof

Instance: i-0908fc43909bba2d Public DNS: ec2-34-245-199-85.eu-west-1.compute.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID	i-0908fc43909bba2d		Public DNS (IPv4)
Instance state	shutting-down		ec2-34-245-199-85.eu-west-1.compute.amazonaws.com
Instance type	m4.10xlarge		IPv4 Public IP
Elastic IPs			34.245.199.85
Availability zone	eu-west-1a		IPv6 IPs
Security groups	launch-wizard-12, view inbound rules		-
Snapshots			Private DNS
			p-172-31-26-53.eu-west-1.compute.internal
			Private IPs
			172.31.26.53
			Secondary private IPs
			VPC ID
			vpc-8133dfe4
			Subnet ID
			subnet-0d778d98
			Network interfaces
			eth0
			Source/dest. check
			True
			T2 Unlimited
			-
			Owner
			328567468974
			Launch time
			March 26, 2018 at 2:21:18 PM UTC+1 (1676 hours)
			Termination protection
			False
			Lifecycle
			spot
			Monitoring
			basic
			Alarm status
			None
			Kernel ID
			-
			RAM disk ID
			-
			Placement group
			hvm
			Virtualization
			Reservation
			r-04133c703dbdd0786

WITH GREAT POWER COMES GREAT VISIBILITY

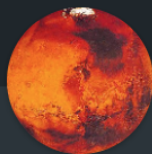
- You control everything. Your research, your way.
- All the power of AWS with none of the late nights learning EC2, networking, SysOps, DevOps, etc..
- Agility to fail fast and recover faster.
- Ability to share and collaborate with a few simple clicks.
- Easily manage your spend with our live budget tools for **guardrails** & **governance**.

RONIN

PROJECT DASHBOARD

MARS PROJECT

< MARS >



DESCRIPTION

For several decades, scientists across the globe have dedicated countless years in pursuit of finding life in Mars, even going as far as planning for humans to migrate to the red planet. And their relentless pursuit may just bear fruit soon, since NASA has already released a detailed plan of how they are going to send humans to Mars in the coming decades. The plan involves sending humans to Mars and have them permanently reside in the planet. According to NASA, "Unlike Apollo, we will be going to stay".

TIMELINE

167

DAYS REMAINING

TAGS

Q DECADES Q SCIENTISTS Q GLOBE

Q DEDICATED Q COUNTLESS Q PURSUIT

Q MARS Q MIGRATE Q RELENTLESS

Q NASA Q DETAILED Q INVOLVES

RONIN

\$5

SPENT

\$847

FORECASTED

\$95

REMAINING

\$100

BUDGET



1

RUNNING

MACHINES



3

STOPPED

MACHINES



\$1.04 P/H

COST

RUNNING MACHINES



97 GB

SSD
STORAGE

\$11.64 MONTH



0 GB

HOT HDD
STORAGE

\$0.00 MONTH



0 GB

COLD HDD
STORAGE

\$0.00 MONTH



0 GB

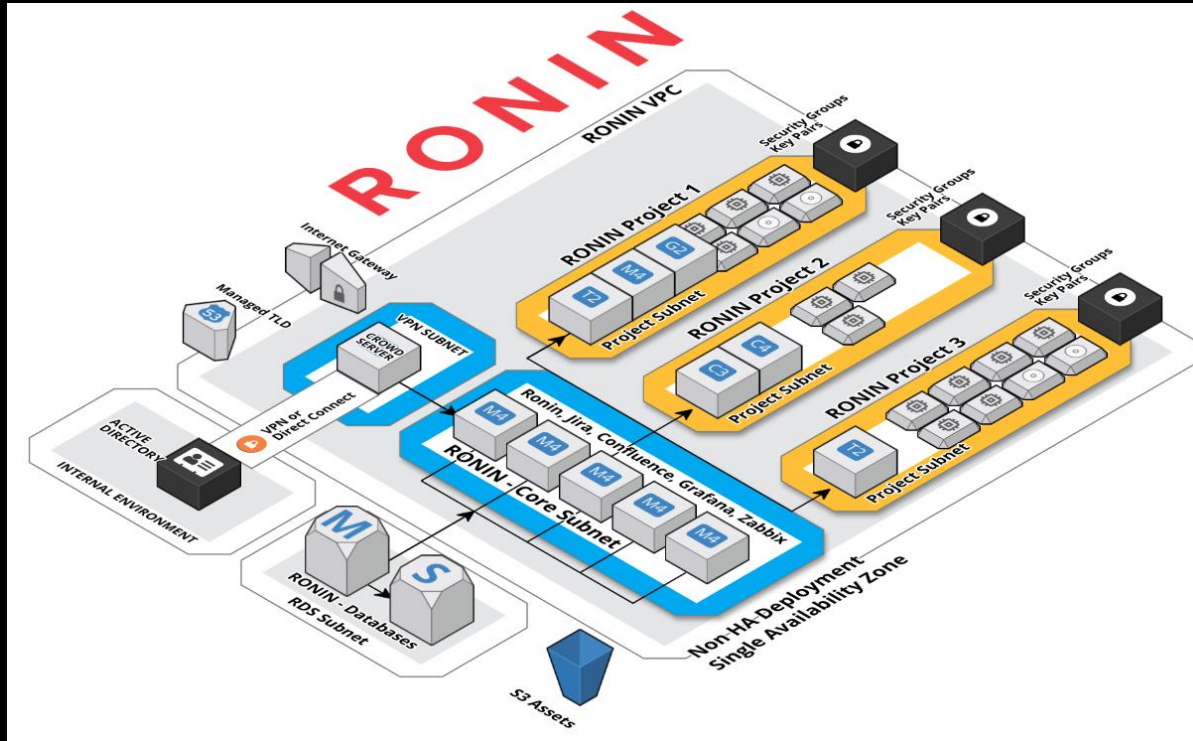
MAGNETIC
STORAGE

\$0.00 MONTH

< 10 MIN >

Boof: DEMO of Ronin

Orchestrating on your behalf



- All activity happens **inside your** own AWS account.
- This means you can **focus once** on your **security and privacy** architecture.
- Ronin will be part of the **enforcement** mechanism.

Nextflow.io

nextflowFeaturesQuick startExamples ▾DocumentationBlogAbout Us

An example

In order to validate Nextflow integration with AWS Batch, we used a simple RNA-Seq pipeline.

This pipeline takes as input a metadata file from the Encode project corresponding to a [search returning all human RNA-seq paired-end datasets](#) (the metadata file has been additionally filtered to retain only data having a SRA ID).

The pipeline automatically downloads the FASTQ files for each sample from the EBI ENA database, it assesses the overall quality of sequencing data using FastQC and then runs [Salmon](#) to perform the quantification over the human transcript sequences. Finally all the QC and quantification outputs are summarised using the [MultiQC](#) tool.

For the sake of this benchmark we used the first 38 samples out of the full 375 samples dataset.

The pipeline was executed both on AWS Batch cloud and in the CRG internal Univa cluster, using [Singularity](#) as containers runtime.

It's worth noting that with the exception of the two configuration changes detailed below, we used exactly the same pipeline implementation at [this GitHub repository](#).

The AWS deploy used the following configuration profile:

```
aws.region = 'eu-west-1'
aws.client.storageEncryption = 'AES256'
process.queue = 'large'
executor.name = 'awsbatch'
executor.awscli = '/home/ec2-user/miniconda/bin/aws'
```

While for the cluster deployment the following configuration was used:

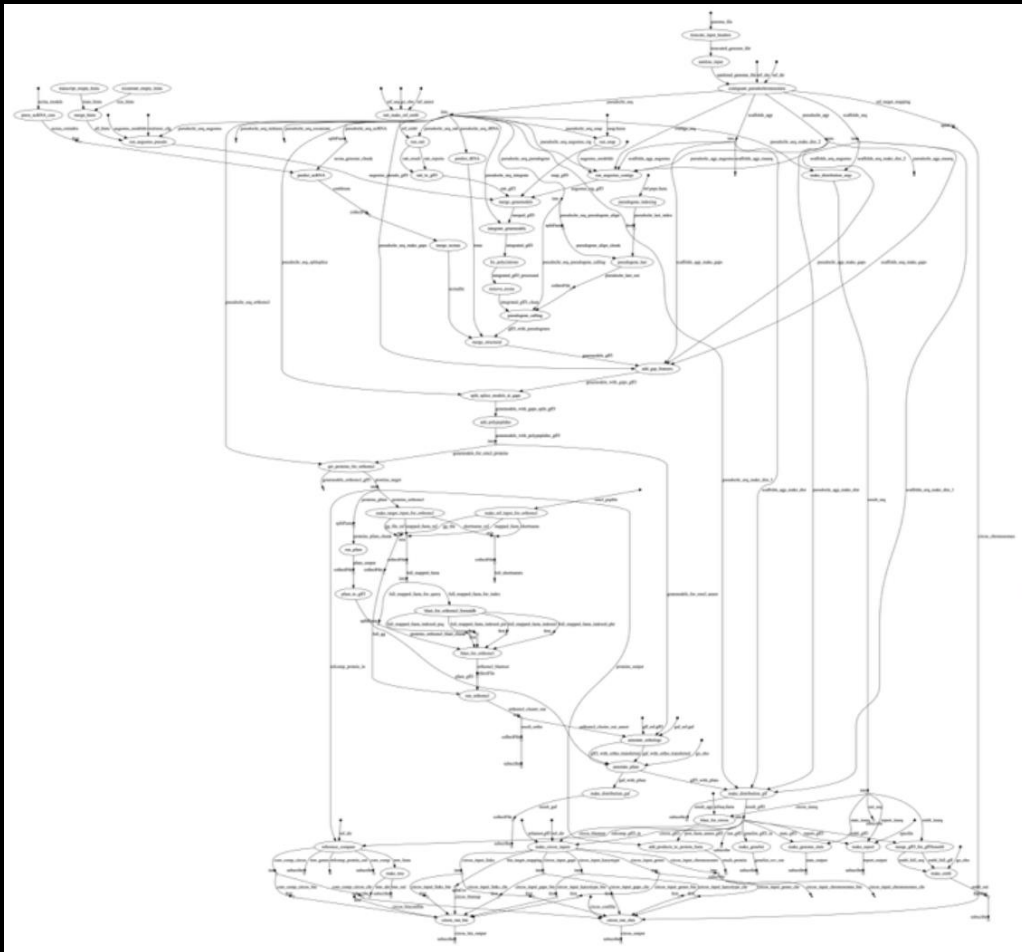
```
executor = 'crg'
singularity.enabled = true
process.container = "docker://nextflow/rnaseq-nf"
process.queue = 'cn-el7'
process.time = '90 min'
process.squant.time = '4.5 h'
```

Nextflow includes built-in support for **AWS Batch**, which that allows the execution of containerised workloads over the Amazon EC2 Elastic Container Service (ECS).

This allows the deployment of Nextflow pipelines in the cloud by offloading the process executions as managed Batch jobs.

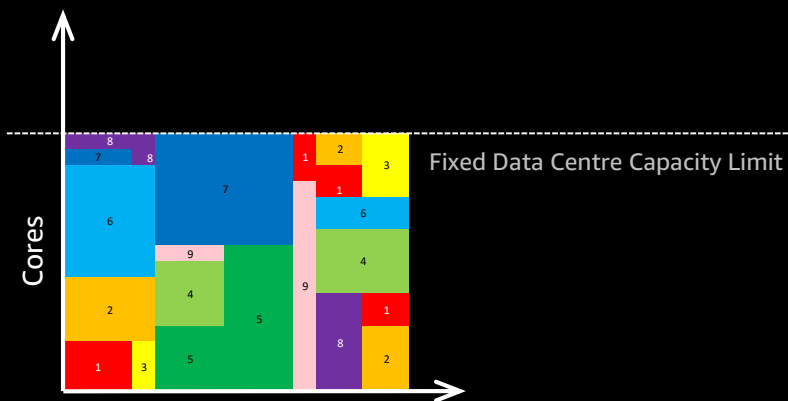
The service takes care to spin up the required computing instances on-demand, scaling up and down the number and composition of the instances to best accommodate the actual workload resource needs at any point in time.





Techniques

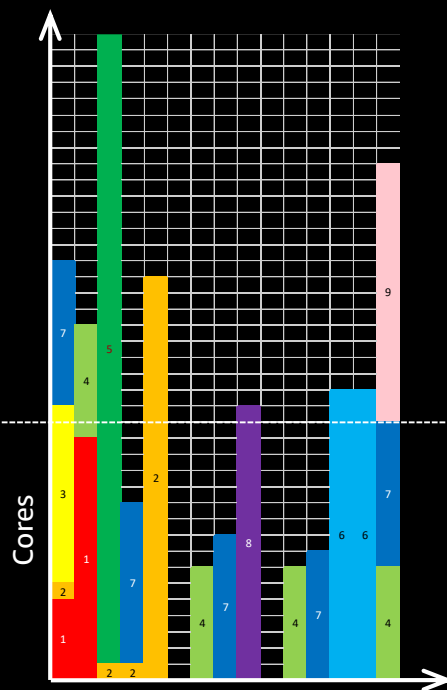
Scaling Research in a Hybrid Cluster Environment



Specialized hardware

Unfortunately finite capacity, usually with long queues to wait in.

Burdened with significant workloads that scale well on AWS.

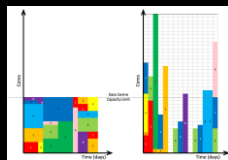


Cloud Expansion Environment

Burst workloads or migrate specific groups to a familiar, almost identical software environment.

Massive capacity when needed to speed up time to results, and agile environment when additional hardware and software experimentation is needed.

Supports and integrates [all major job schedulers](#).



Cores

Time (days)

Pop-Up Compute Clusters – Research Tech

Introducing **Alces Flight** - self-scaling HPC-style clusters instantly ready to compute, billed by the hour and using the AWS Spot market by default to achieve supercomputing for ~1c per core per hour.



<http://alces-flight.com/>

- 1,500+ popular scientific applications
 - Pre-installed & ready to run.
- Available via **AWS Marketplace** (the cloud's "App Store") and launched within minutes.
- Exploits Amazon **EC2 Spot market** by default.
- Deployable anywhere on Earth ... **immediately**.

Elastic HPC Cluster



< 10 MIN >

Ralph: DEMO of Alces Flight + OpenFOAM

Clusters as code



```
$ generate_data | filter1 --cpus | filter2 --gpus
```

Take aways

- Focus on the things slowing down the flywheel.
- **Humans need the most help right now.**
- Make software a bit better, and then a bit better again:
 - **Automate** everything (humans suck at repetition)
 - Unburden people of the **crap tasks**.
- Tell us how we can help. **Don't be shy**.



<http://boofla.io/ronin101>



alcesflight

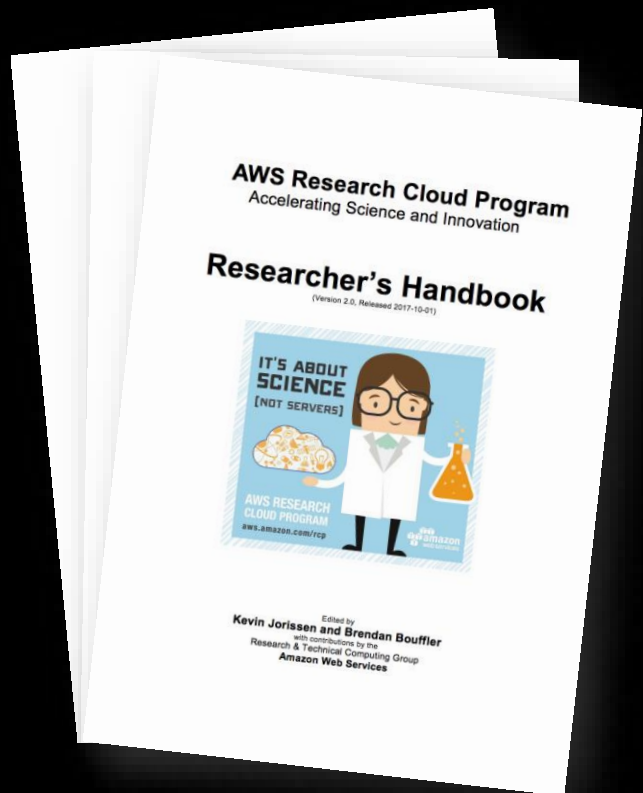
<http://alces-flight.com/>



QUESTIONS?

AWS Researcher's Handbook

The 200-page “**missing manual**” for science in the cloud.



Written by Amazon's Research Computing community **for scientists**.

- **Explains** foundational concepts about how AWS can accelerate time-to-science in the cloud.
- **Step-by-step best practices** for securing your environment to ensure your research data is safe and your privacy is protected.
- **Tools for budget management** that will help you control your spending and limit costs (and preventing any over-runs).
- **Catalogue of scientific solutions** from partners chosen for their outstanding work with scientists.

aws.amazon.com/rcp

