

CAKE

Walkthrough

- 1 min → Introduction
- 4 min → Section I – Survey
- 3 min → Section II – Proof of concept of CAKE
- 2 min → Section III – Research Notes and Conclusion
- 1 min → Question and Answers



- this symbol in slide indicates our contributions

Introduction

Cluster Allocation for Kernel Execution using AI (**CAKE**)

- what it is ?
 - It is a **pluggable** scheduler framework for better scheduling decisions.
 - **Experimental.**

Keyterm - Scheduling

Scheduler Primary Constraint : Minimize **job latency** while maximizing cluster utilization

Key Challenges in Scheduling

Scale **Heterogeneous workload** Maximize utilization

Little more details

- **Cluster** – Homogenous / Heterogeneous computers
- **Allocation** – Scheduling. So run Job J1 in XYZ machine.
- **Kernel** – Any code of a Job which requires data/compute nodes.
- **Execution** – Right decision on Right Machine
- **AI**
 - Used Genetic Algorithm to model the scheduling.
 - Also some research notes on existing work

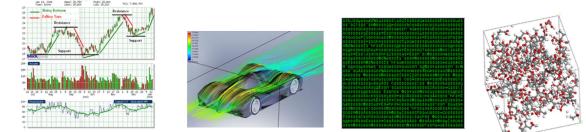


image credit - <http://brainz.org/15-real-world-applications-genetic-algorithms/>



Monolithic scheduler. Two-level scheduling. Shared-stat e scheduling. Distributed scheduling. Hybrid scheduling.

Borg

Mesos

Omega

Sparrow

Hawk

Quasar

YARN

Apollo

Tarcil

Paragon

Nomad

Mercury

Other Important papers

Genetic Algorithm

A Review of Machine Learning in Scheduling

Prediction of stability of the clusters in MANET using

MR-GABiT: Map Reduce based Genetic Algorithm

Planning consideration in HPC Clusters

The evolution of cluster scheduler architectures

Cooperative Batch Scheduling for HPC Systems

I/O-Aware Batch Scheduling for Petascale Computing Systems

Exploring Plan-Based Scheduling for Large-Scale Computing Systems

7

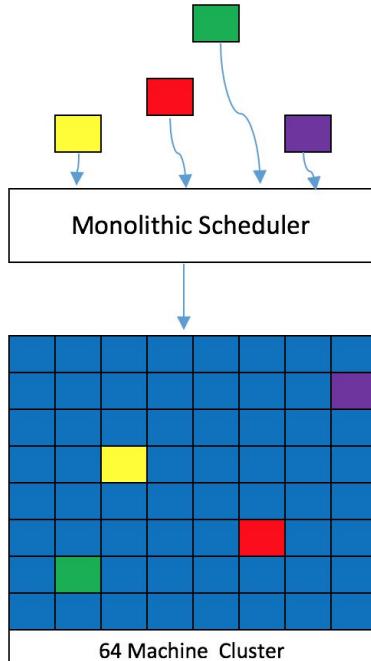
A Data Driven Scheduling Approach for Power Management on HPC Systems

[More info - <http://www.cs.iit.edu/~zlan/spear.html>]



Many Cluster Scheduler : Monolithic Scheduler

+++ Simple and Uniform



Examples -

- Borg
- Hadoop Schedulers
- Kubernetes
Schedulers

Google's Borg System

Applications

Web Search

Gmail

Google Docs

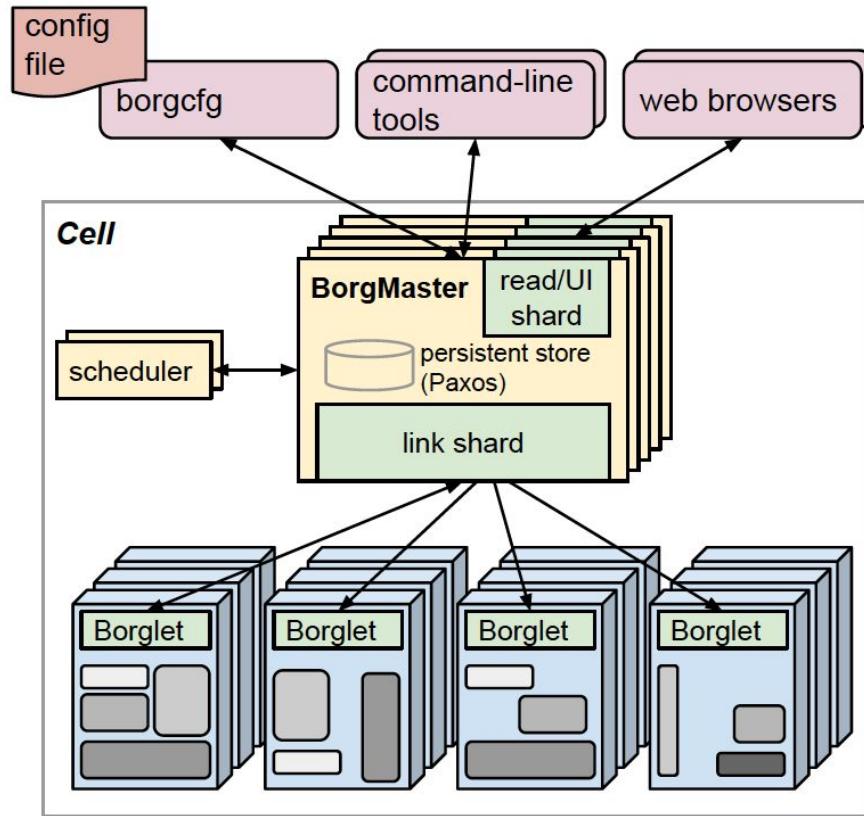
Distributed File System

GFS

CFS

Bigtable

Megastore



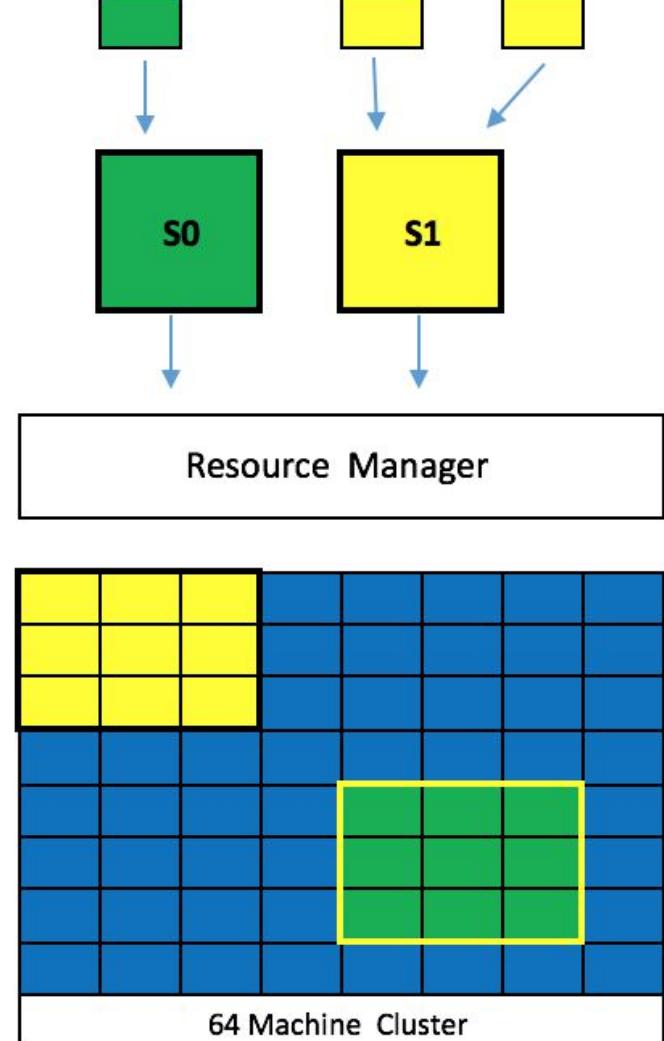
Scheduling in Borg

- Step 1:
 - Finding Feasible nodes
- Step 2:
 - Scoring

Figure 1: The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*

Two Level Schedulers

Scheduler Offers resources to another scheduler.



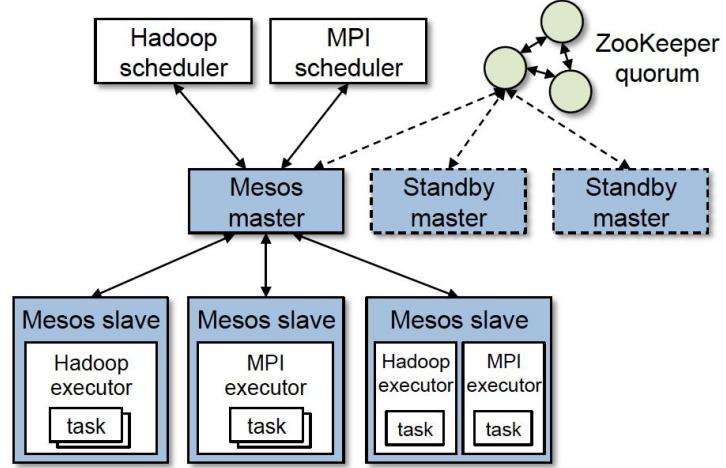


Figure 2: Mesos architecture diagram, showing two running frameworks (Hadoop and MPI).

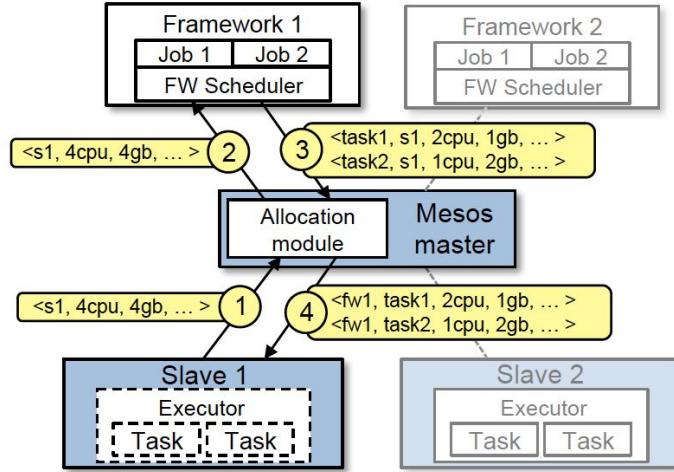


Figure 3: Resource offer example.

MESOS

YARN

- Uses following Scheduling policies
 - FIFO
 - FAIR
 - Dominant Resource Fairness

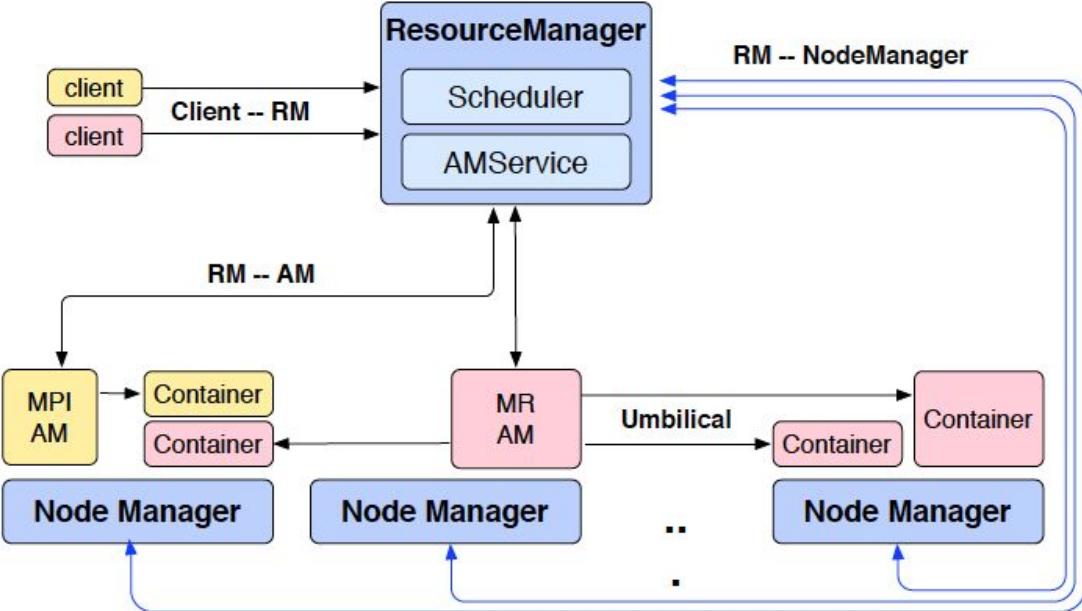


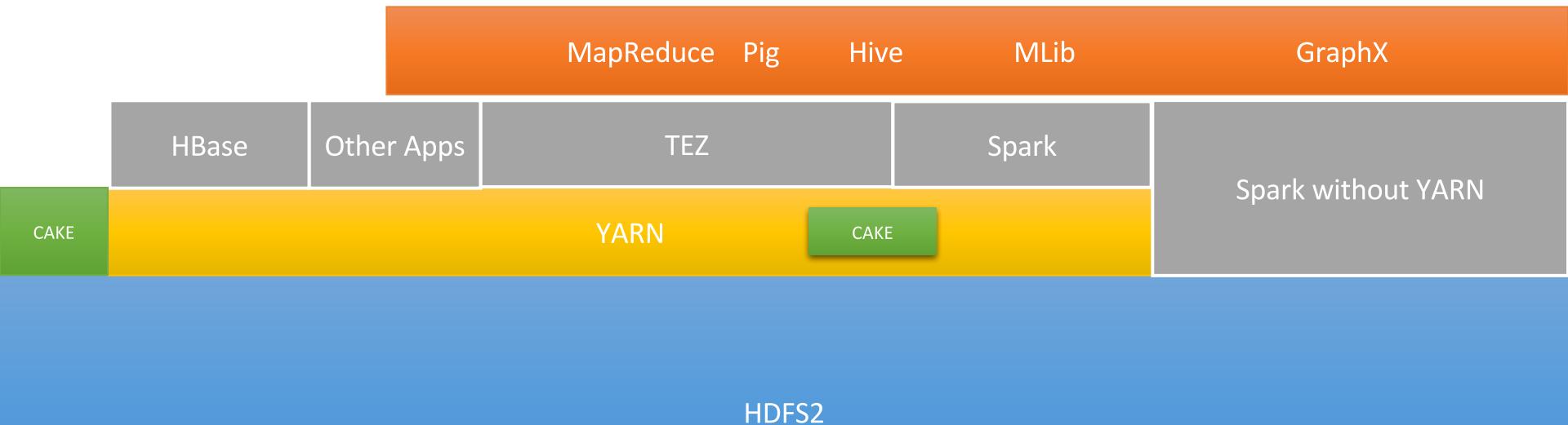
Figure 1: YARN Architecture (in blue the system components, and in yellow and pink two applications running.)

What is **not** easy in two-level scheduling ?

Application-specific schedulers care about many different aspects of the underlying resources, but their only means of choosing resources is the **offer/request interface** with the resource manager. This interface can easily become quite complex

many more... *discussed in report*

Where it fit in technology stack?

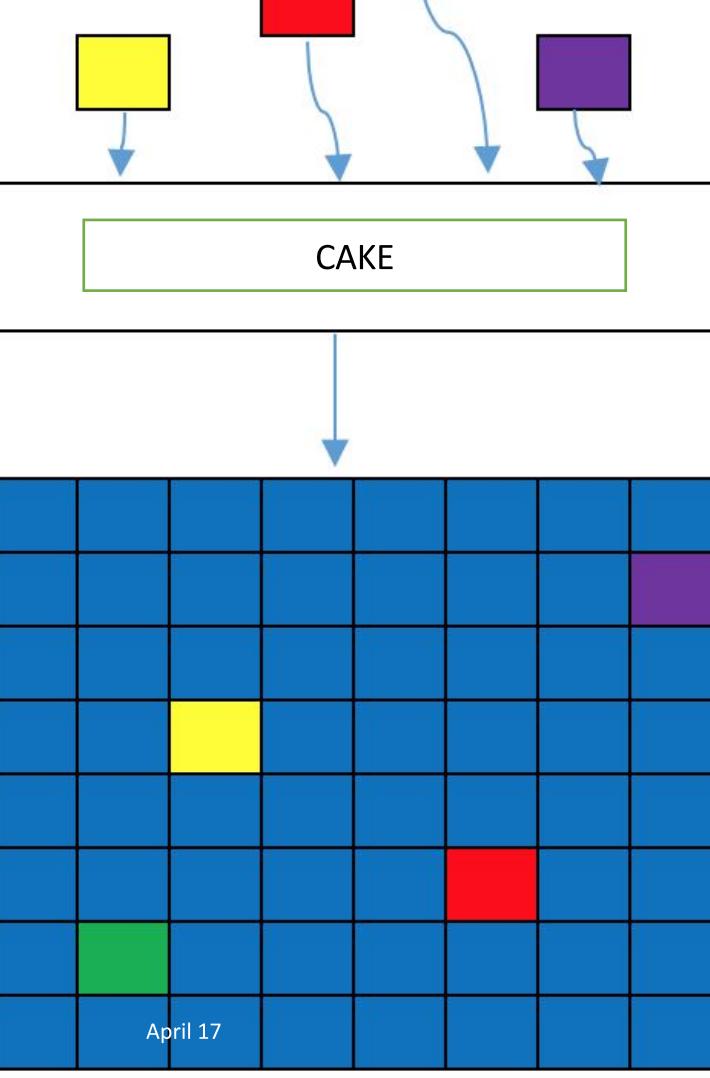


Section II

– Proof of concept

CAKE

(Goal, Design, Flow)



- ◆ Goal

Framework Design Goals

1. Adaptability to existing Schedulers
2. Adapts to Scale (10 Jobs/sec to 1million Jobs/sec)
3. Extensibility
4. Easy to use...
5. ...





CAKE webservice

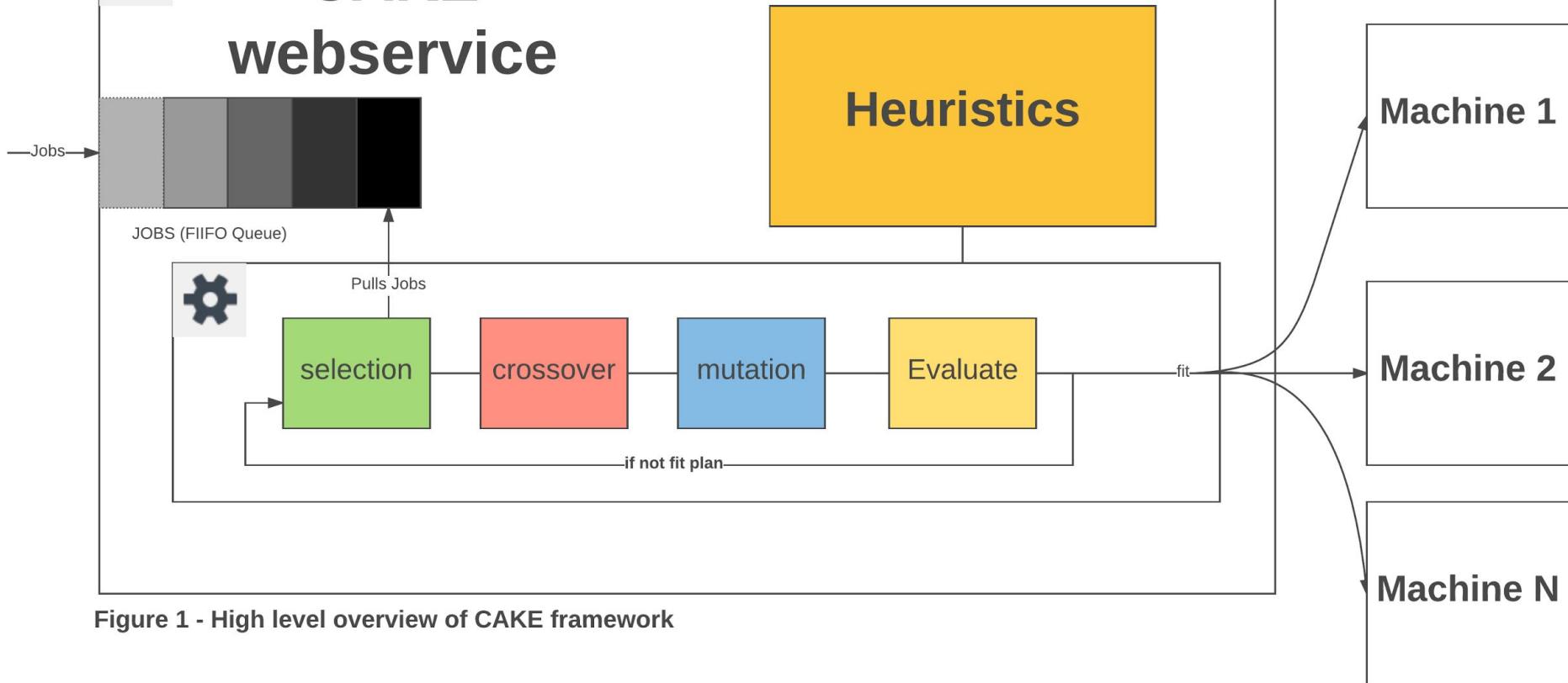


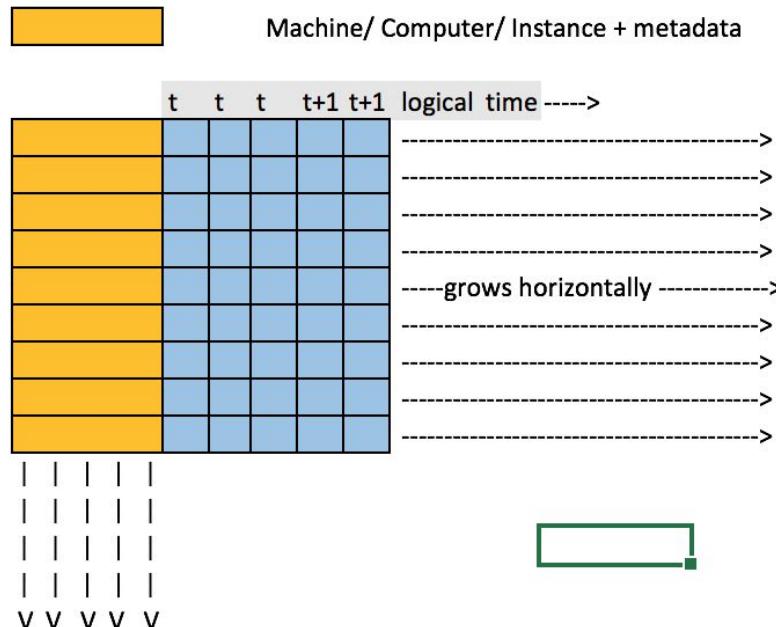
Figure 1 - High level overview of CAKE framework



Output

April 17

A Single Plan = Machines x Jobs Matrix



 the Job has application level constraints



test case 1	more jobs	75
--------------------	-----------	----

Allocation is based on
Sequential Order

CAKE Plan 1 for 75 Jobs

Machine 1	1	9	16	24	32	40	48	56	64	71
Machine 2	2	10	17	25	33	41	49	57	65	72
Machine 3	3	11	18	26	34	42	50	58	66	73
Machine 4	4	12	19	27	35	43	51	59	67	74
Machine 5	5	13	20	28	36	44	52	60	68	75
Machine 6	6	14	21	29	37	45	53	61	69	
Machine 7	7	15	22	30	38	46	54	62	70	
Machine 8	8	16	23	31	39	47	55	63	71	

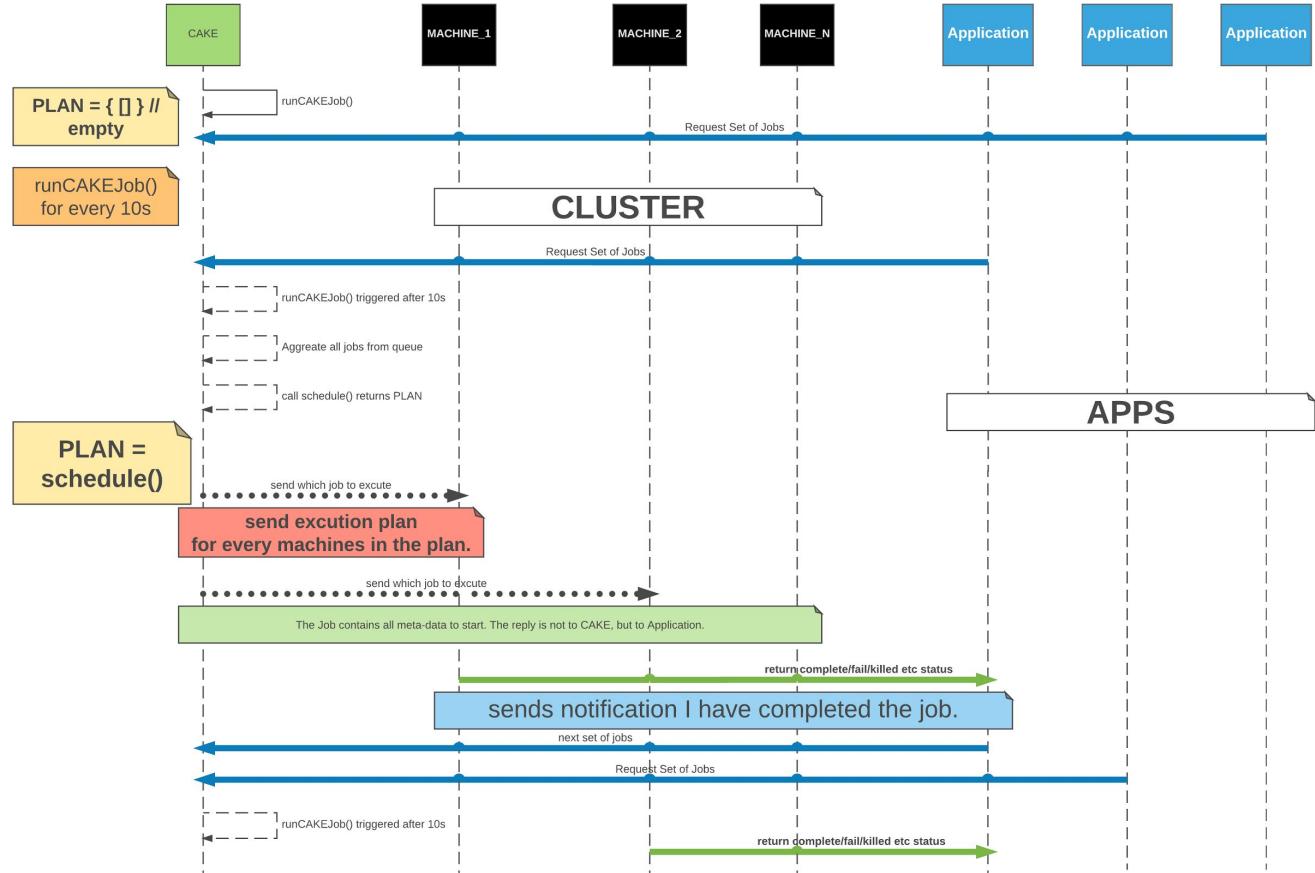
→ Means in Machine 1, Jobs having ID 1,9, 16, 24, 32, 40, 48, 56, 64, 71 will be executed

Example

High Level Sequence Diagram

Execution Flow

- Quite Simple.
- Future work
 - Sequence Diagram for real-world application

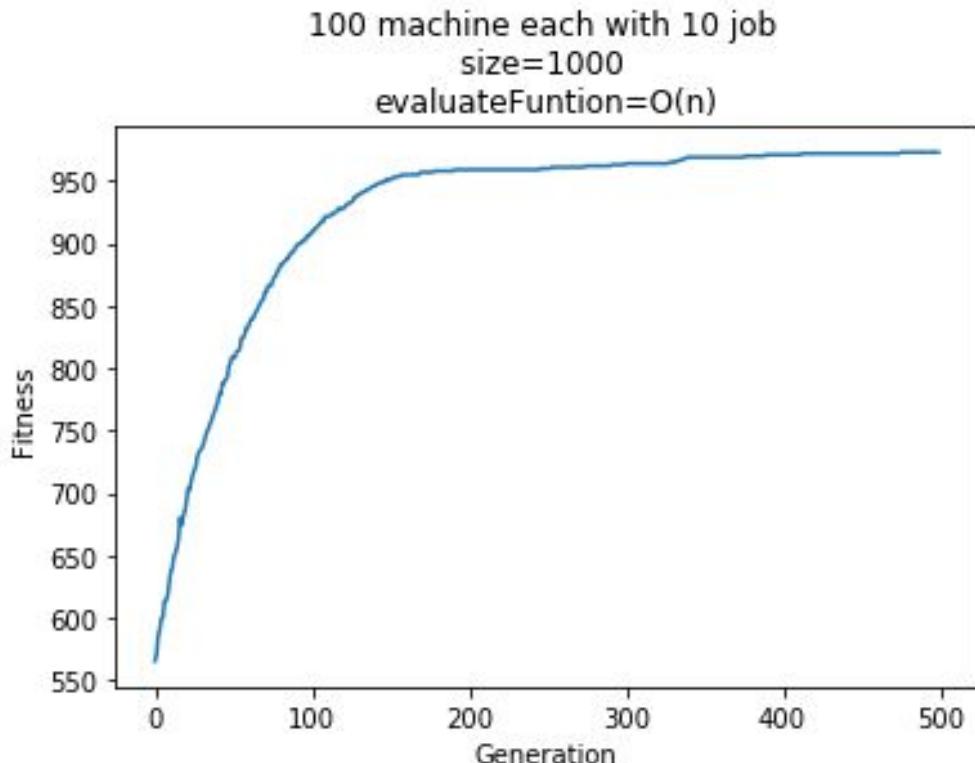


- ◆ Section III
- Research Notes and Conclusion
(Results, Critics, Big picture)

Results ◆

- We wrote GA in Python.
- Not a good Implementation.
 - Serial implementation.
 - Not optimized code.
- For Production- we need to implement it in C++ by leveraging GPU using CUDA.
- Evaluate(schedule, machine,jobs){
 - Check if all the jobs are scheduled.
 - Check if total requirements by all the jobs scheduled on the machine are not exceeding available resources on the machine.
 - Return fitness
- **More analysis & Graphs is in report.**

search : i-08a7cd57085bdb84f			Add filter
Name	Instance ID	Instance Type	
m4.2xlarge-deleteitnow	i-08a7cd57085bdb84f	m4.2xlarge	



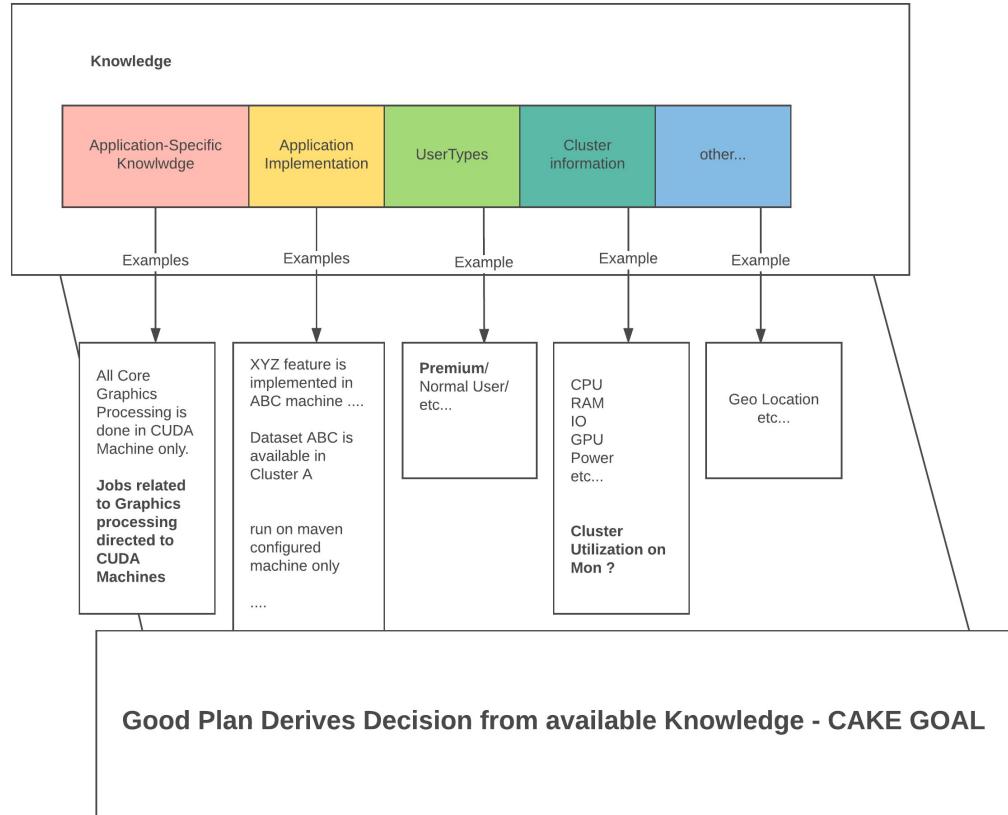
Critics

- Lot of engineering work
- Latency depends on ***how and where*** CAKE is implemented.
- Complex to write heuristics for evaluate functions
- ...
- ... *written in report*

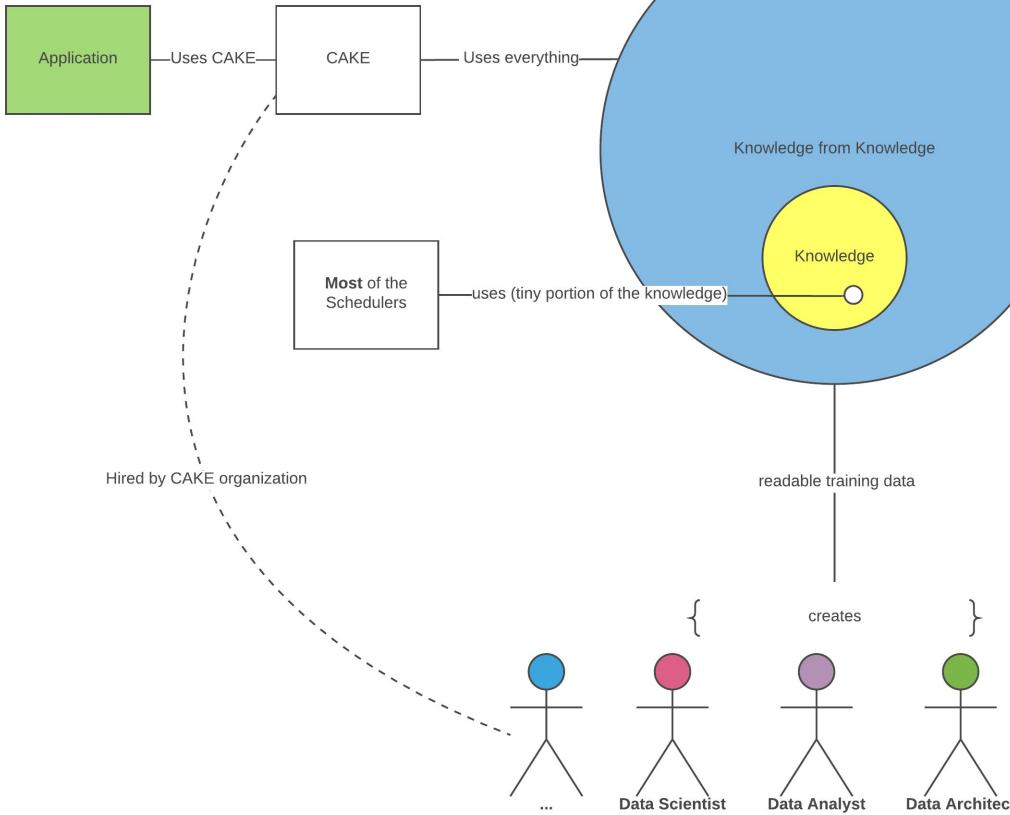


Knowledge has beginning but no end.

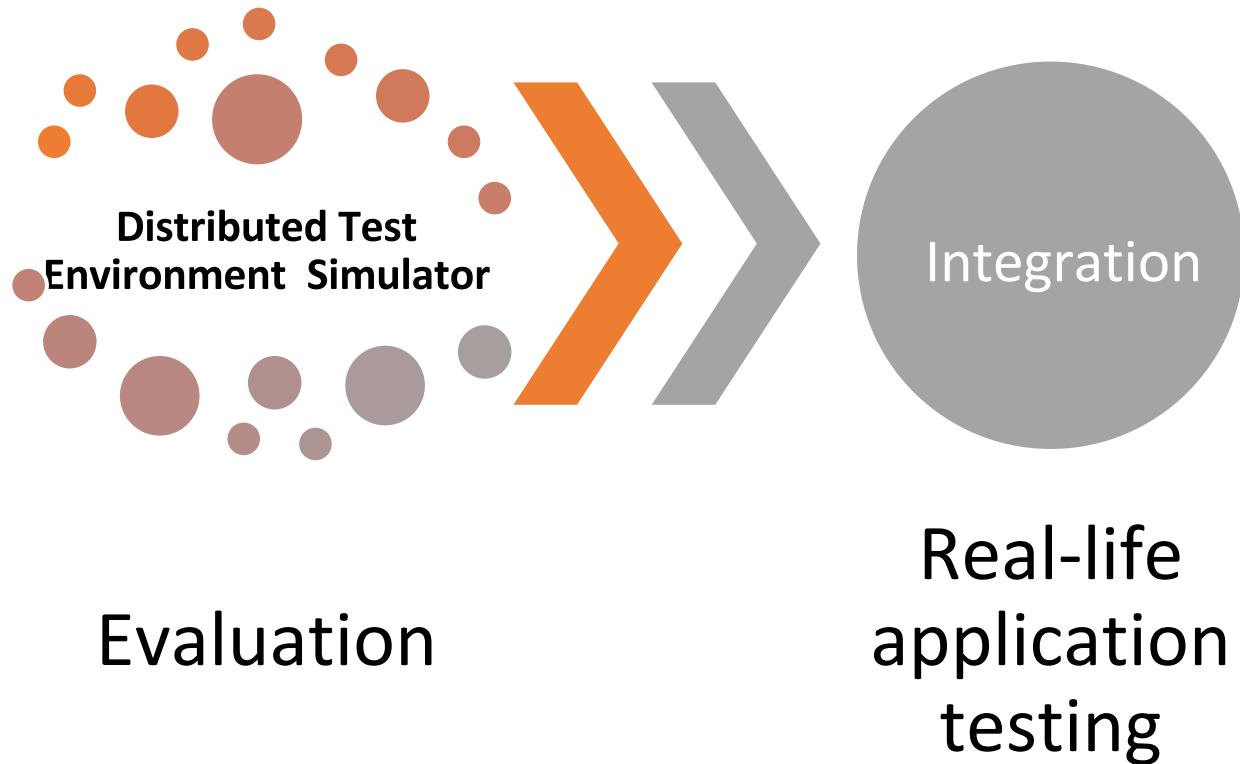
April 17



Big Picture



FUTURE WORK



How does CAKE differ from BORG ?

It differs, but it has same underlying design principle , that is scheduling based on scoring.. ***we are aiming to take CAKE framework to be a superset of BORG.***

CAKE is a framework similar to AngularJS, Spring Framework etc... but this is for Scheduling....

CAKE is not **Kubernetes** scheduler, it differs – discussed in report.



SHARE



SHARE
3992



TWEET



COMMENT
11



EMAIL

CADE METZ BUSINESS 06.10.14 1:15 PM

GOOGLE OPEN SOURCES ITS SECRET WEAPON IN CLOUD COMPUTING



SHARE



SHARE
3992



TWEET



COMMENT
11



EMAIL

Illustration: Getty

WHEN GOOGLE ENGINEERS John Sirois, Travis Crawford, and Bill Farner left the internet giant and went to work for Twitter, they missed Borg.

Borg was the sweeping software system that managed the thousands of computer servers underpinning Google's online empire. With Borg, Google engineers could instantly grab enormous amounts of computing power from across the company's data centers and apply it to whatever they were building—whether it was Google Search or Gmail or Google Maps. As Sirois, Crawford, and Farner created new web services at Twitter, they longed for the convenience of this massive computing engine.

Unfortunately, Borg was one of those creations Google was loath to share with the outside world—a technological trade

<https://www.wired.com/2013/03/google-borg-twitter-mesos/all/>

Why was Borg Closed source?

...

Artificial Intelligence is the New Electricity - Andrew Ng

Conclusion ...

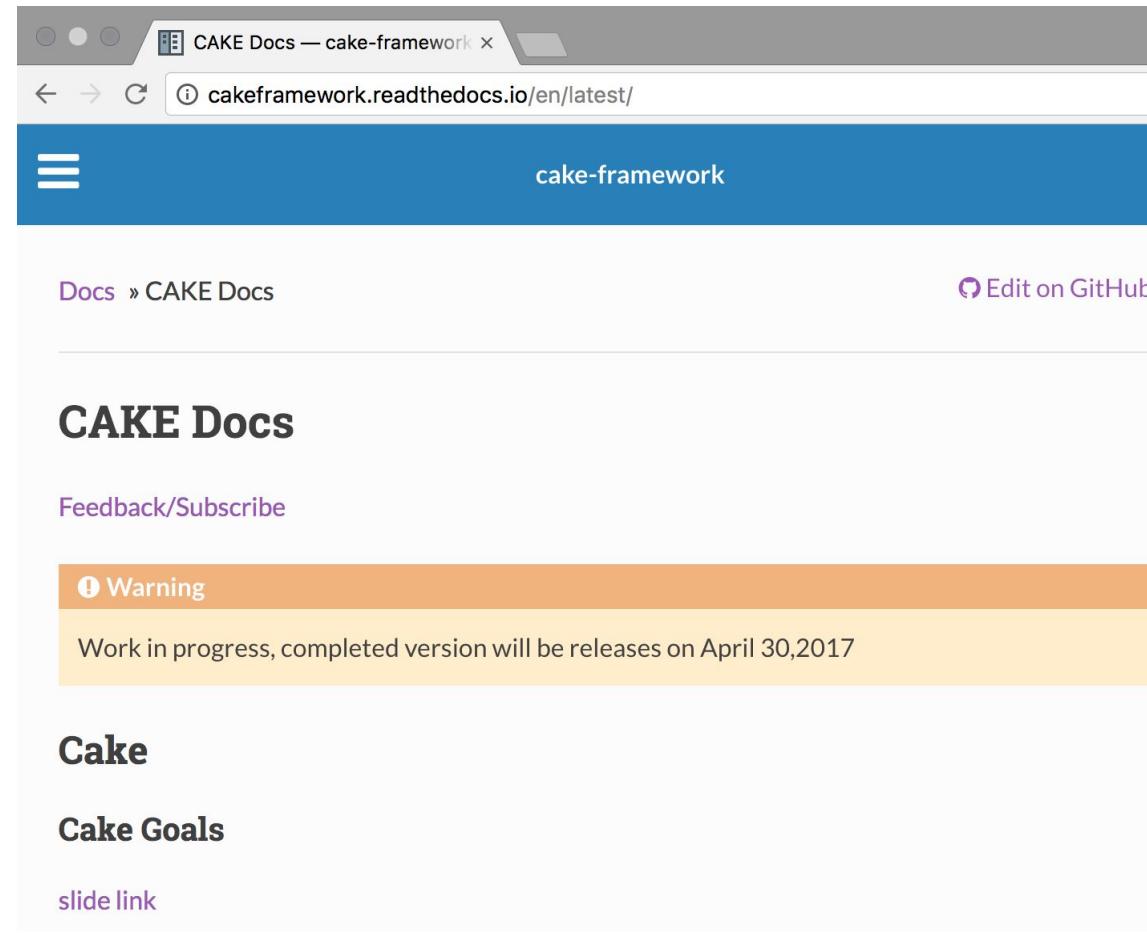
Advance Concepts(AI,etc) in Scheduling must be unified by all **researchers** in Apache Open Source project, such that any application or organization will not worry about scheduling at all.

CAKE project is our attempt ...

Thank you

-Ajay Ramesh

-Chandra Kumar



A screenshot of a web browser displaying the CAKE Docs page for the cake-framework project. The page has a blue header with the title "cake-framework". Below the header, there's a breadcrumb navigation "Docs » CAKE Docs" and a purple "Edit on GitHub" button. The main content area features a large heading "CAKE Docs" and a "Feedback/Subscribe" link. A yellow warning bar at the bottom left contains the text "Warning: Work in progress, completed version will be released on April 30, 2017". The URL in the address bar is "cakeframework.readthedocs.io/en/latest/".

CAKE Docs — cake-framework

cake-framework

Docs » CAKE Docs

Edit on GitHub

CAKE Docs

Feedback/Subscribe

Warning

Work in progress, completed version will be released on April 30, 2017

Cake

Cake Goals

slide link