Towards Enabling Big Data and Federated Computing in the Cloud

Yousef Kowsar[1], Enis Afgan[1,2]
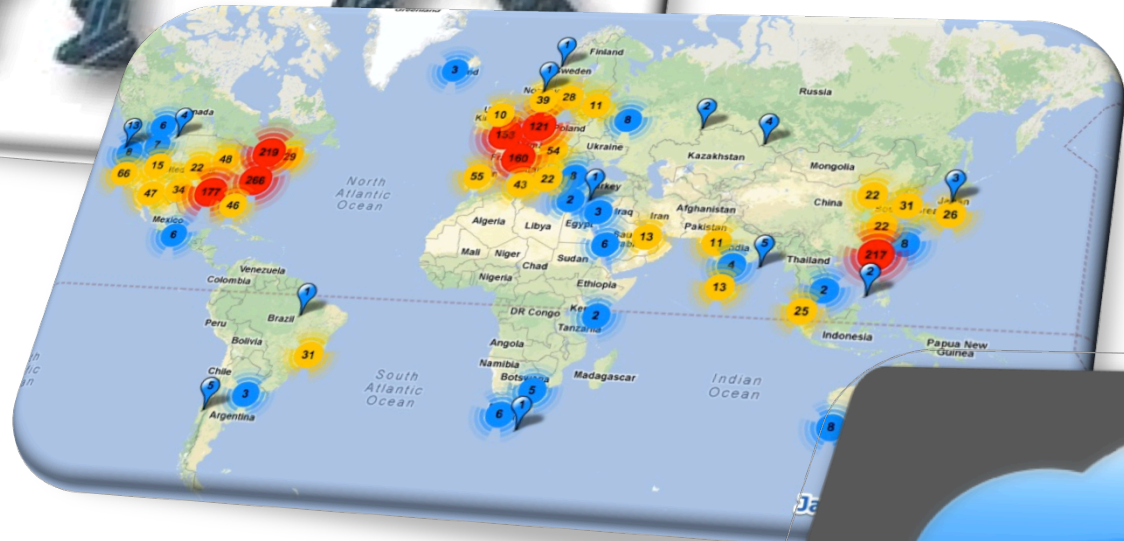
[1]VLSCI, University of Melbourne

[2]CIR, Ruđer Bošković Institute (RBI)
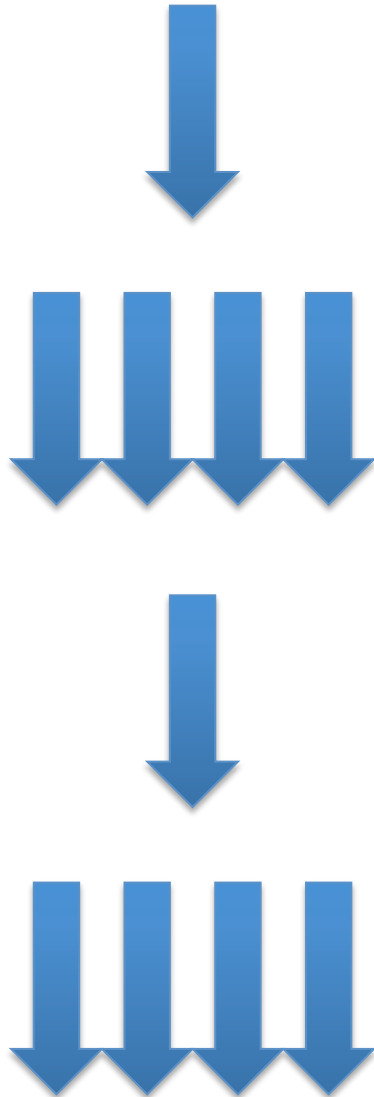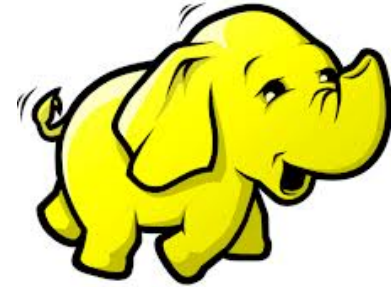
BOSC 2013, Berlin

CloudMan

Lots of big data

Distributed data

'Cloud' resources

Big Data and Hadoop

Prominent users [edit]

Yahoo! [edit]

On February 19, 2008, Yahoo! Inc. launched what it claimed was the world's largest Hadoop production application. The Yahoo! Search Webmap is a Ha Linux cluster and produces data that is used in every Yahoo! Web search query.[29]

There are multiple Hadoop clusters at Yahoo! and no HDFS filesystems or MapReduce jobs are split across multiple datacenters. Every Hadoop cluster distribution. Work that the clusters perform is known to include the index calculations for the Yahoo! search engine.

On June 10, 2009, Yahoo! made the source code of the version of Hadoop it runs in production available to the public.[30] Yahoo! contributes back all w company's developers also fix bugs and provide stability improvements internally, and release this patched source code so that other users may benefi

Facebook [edit]

In 2010 Facebook claimed that they had the largest Hadoop cluster in the world with 21 PB of storage.[31] On July 27, 2011 they announced the data ha the data had grown to 100 PB.[33] On November 8, 2012 they announced the warehouse grows by roughly half a PB per day.[34]

Other users [edit]

Besides Facebook and Yahoo!, many other organizations are using Hadoop to run large distributed computations. Some of the notable users include:[5]

- Amazon.com
- Ancestry.com[35]
- Akamai
- American Airlines
- AOL
- Apple[36]
- AVG
- eBay
- Electronic Arts
- Ericsson
- Hortonworks
- Federal Reserve Board of Governors
- Foursquare
- Fox Interactive Media
- Google
- Hewlett-Packard
- IBM
- ImageShack
- ISI

- InMobi[37]
- Intuit
- Joost
- Last.fm
- LinkedIn[38]
- Microsoft[39]
- NetApp
- Netflix[40]
- Ooyala
- Riot Games
- Spotify
- Qualtrics
- US National Security Agency (NSA)
- The New York Times
- SAP AG[41]
- SAS Institute[42]
- StumbleUpon[43]
- Twitter
- Yodlee

Federated Computing and HTCondor

- An approach toward federated computing
- HTCondor:
  - Since 1988 at University of Wisconsin-Madison
  - High Throughput Computing on large collections distributive computing resources: *cycle scavenging*
- Gains from using HTCondor
  - Existing solution
  - Scalability
  - Reliability
  - Cost
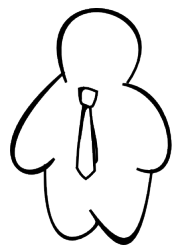
CloudMan

- Cloud Manager for orchestrating cloud resources

- Cluster-on-the-cloud, any cloud

- Ease the process of establishing a cloud environment for bioinformatics analysis
  - "Galaxy on the Cloud"

- Facilitate management of IaaS services

A path forward

- Have a central manager capturing all the three functions at once:
  - CloudMan
    - Easy & ready to use cluster environment for the cloud
  - Hadoop
    - Platform for Big Data analysis
  - HTCondor
    - Central manager able to handle versatile, heterogeneous compute environments

Our Approach
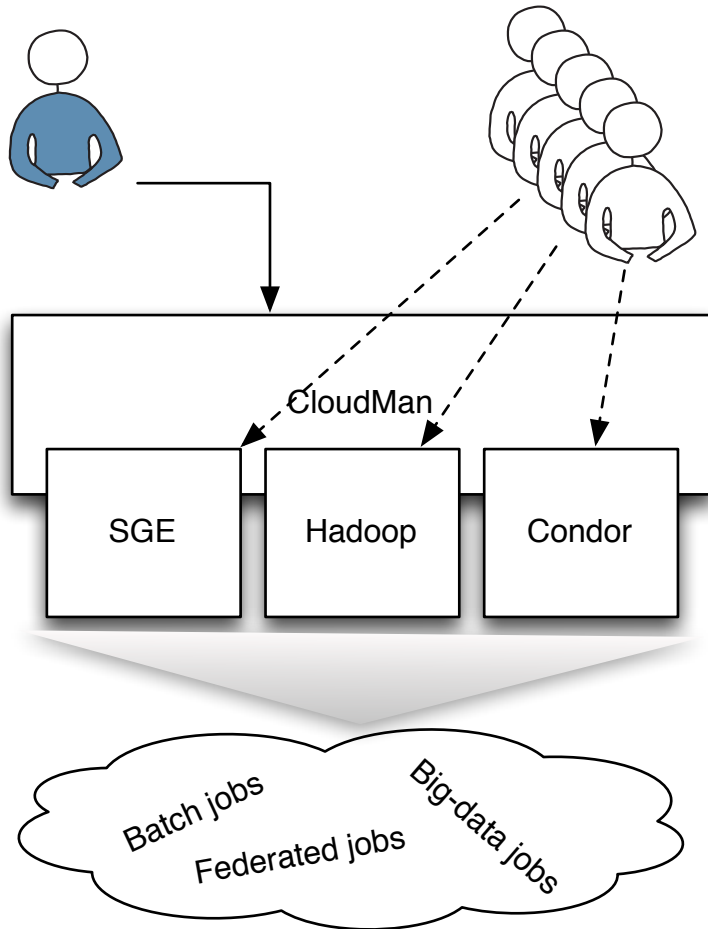
- Integrate HTCondor and Hadoop into CloudMan clusters
- Single management interface
- Multiple types of workloads and infrastructures

- Make it easier to deploy necessary platform and enable
  1. Tool development
  2. Data analysis

Hadoop-on-demand platform
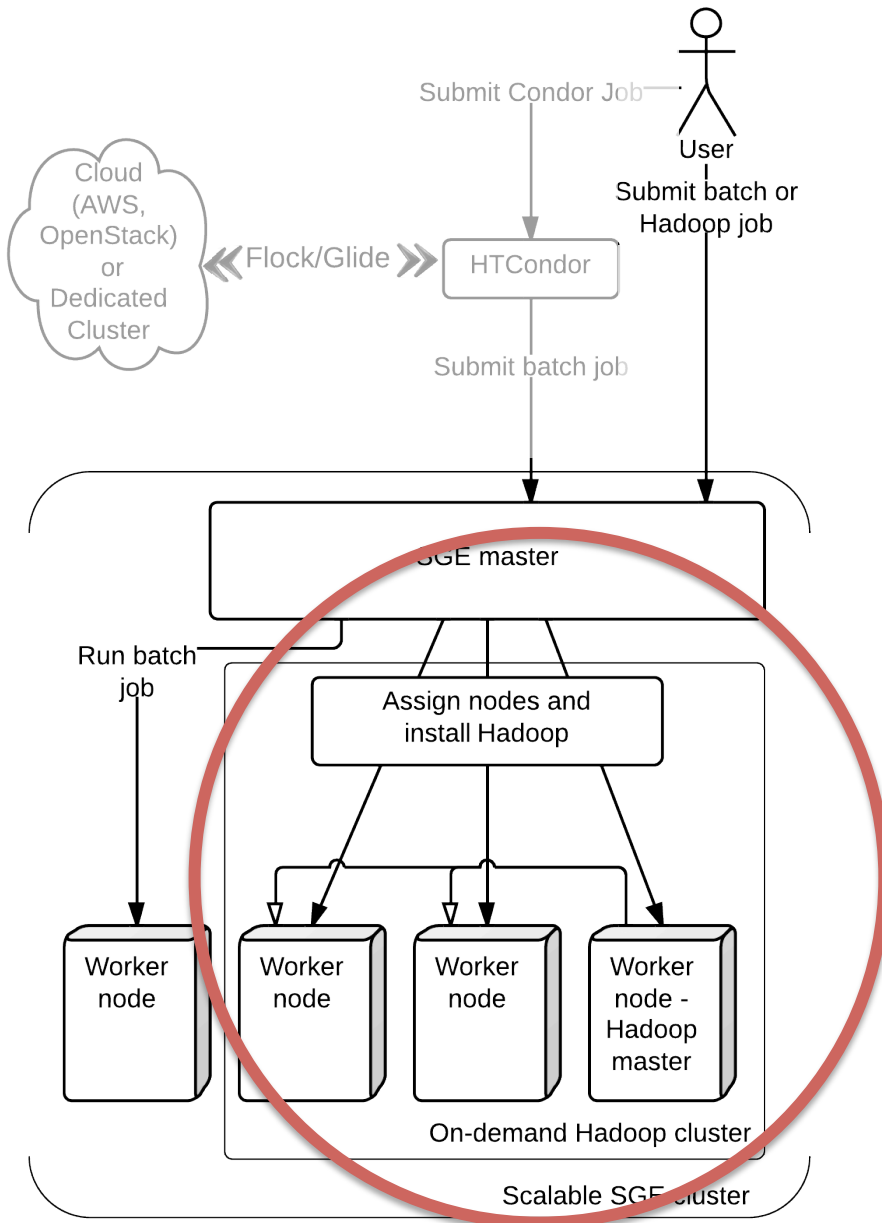
- *Hadoop-over-SGE*: dynamically setup at runtime
- Low and constant setup overhead
- Increase infrastructure flexibility
  - Cost
  - Workload type

System cleanup (2s)

Job runtime (variable)

Hadoop setup (114s)

Job handover (<1s)

Hadoop preparation (118s)

SGE allocation (~13s)
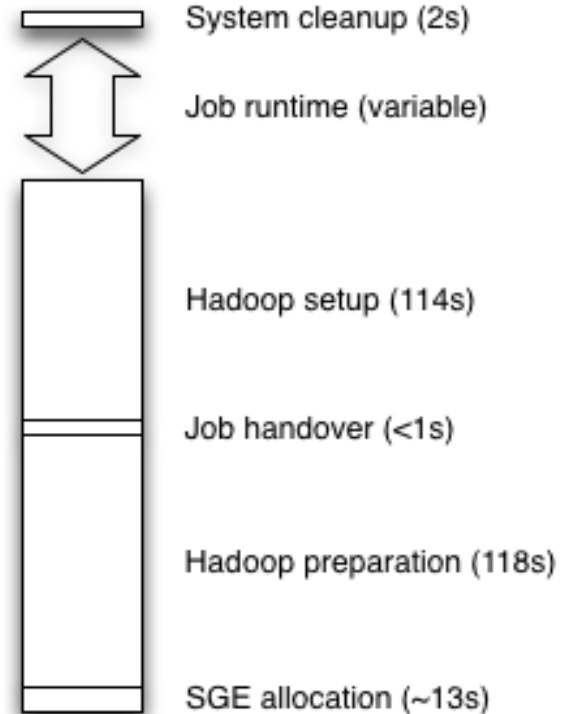
Hadoop example

- Edit sge-integration script


```
#!/bin/sh

## submit this script using the command below for example
## qsub -v HADOOP_HOME=$HOME/hadoop-1.0.2,JAVA_HOME=/usr hdfs-sge.cmd
##

#$ -N hadoop
#$ -o /tmp/out.$JOB_ID
#$ -j y
#   change the number from 16 to 32 (4 nodes) or 64 (8 nodes)
#$ -pe mpi 2          ——— Change this as required
#$ -cwd
#   No need to change this line, this line tells SGE to run jobs in hadoop.q
#
```

```
                                                Set your Hadoop Commands
                    mapred.sh
sleep 10
$HADOOP_HOME/bin/hadoop fs -put /home/ubuntu/hadoop/hadoop/home/conf input
echo "input file copied sleeping for 30 seconds"
sleep 30
$HADOOP_HOME/bin/hadoop jar hadoop-examples-1.0.4.jar  wordcount input output
$HADOOP_HOME/bin/hadoop fs -get output $HADOOP_HOME/output.$JOB_ID
$HADOOP_HOME/bin/stop-mapred.sh
$HADOOP_HOME/bin/stop-dfs.sh
```

- Submit your job into SGE

```
:~$ qsub -v HADOOP_HOME=/home/ubutnu,JAVA_HOME=/user hdfs-sge.cmd
```

HTCondor integration

- Local jobs run via SGE
- Nodes pooled together via
  - Flocking
  - Gliding
  - Pool sharing

Cluster

NeCTAR
(~private)

Campus

AWS
$$$

Submit Condor Job

User

Submit batch or
Hadoop job

Cloud
(AWS,
OpenStack)
or
Dedicated
Cluster

≪ Flock/Glide ≫

HTCondor

Submit batch job

SGE master

Run batch
job

Assign nodes and
install Hadoop

Worker
node

Worker
node

Worker
node

Worker
node -
Hadoop
master

On-demand Hadoop cluster

Scalable SGE cluster
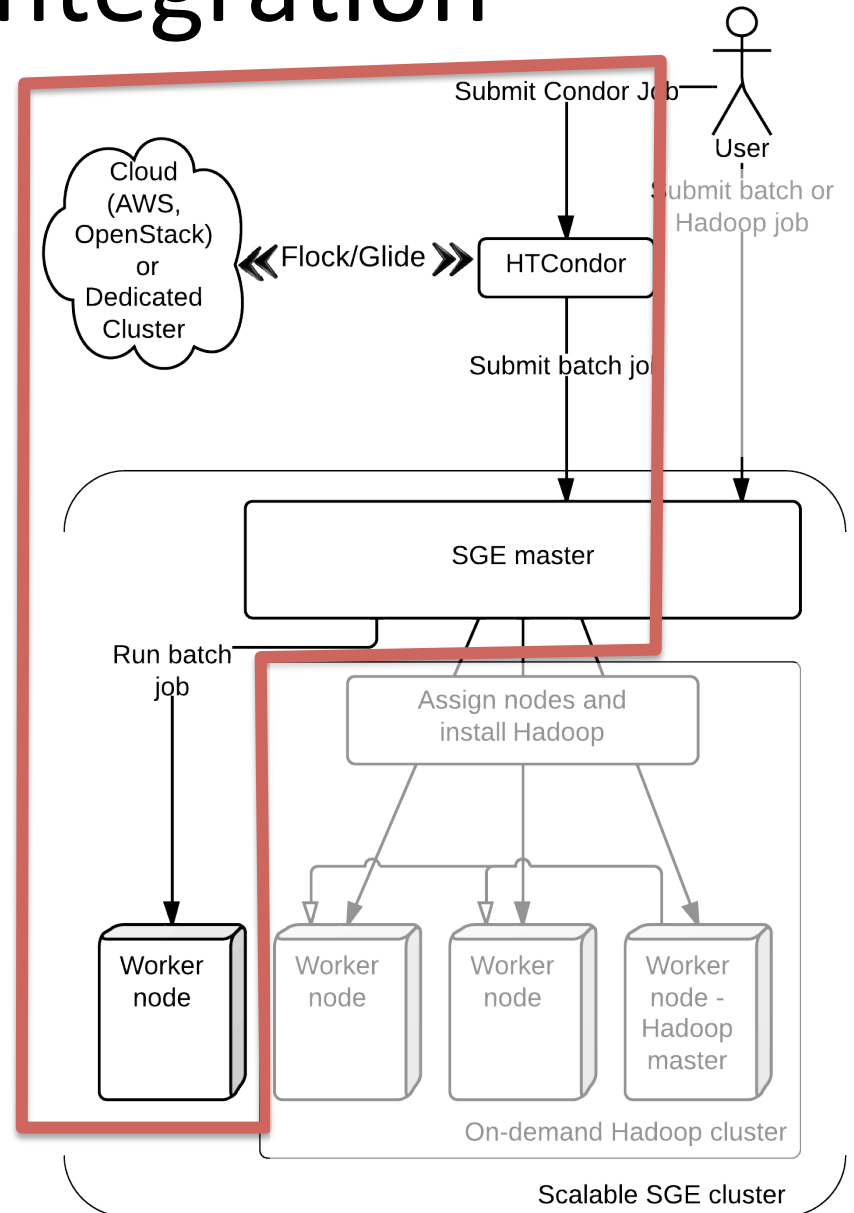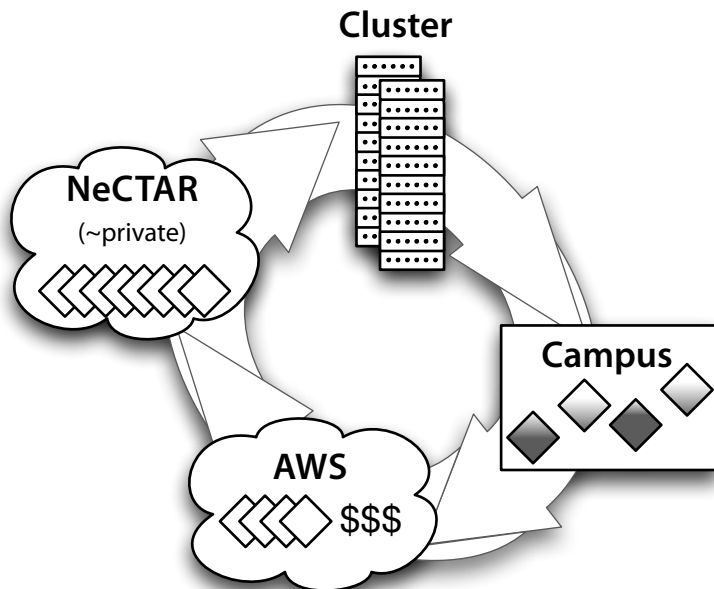
HTCondor example

Cluster 1 - AWS

CloudMan Console

Welcome to CloudMan. This application allows you to manage this instance cloud cluster and the services provided within. Your previous data store has been reconnected. Once the cluster has initialized, use the controls below to manage services provided by the application.

| Terminate cluster | Add nodes ▾ | Remove nodes ▾ | Access Galaxy |

Status

Cluster name: cndr_f3
Disk status: 0 / 0 (0%)
Worker status: Idle: 1 Available: 1 Requested: 1
Service status: Applications ● Data ●

Master Node
i-00005566
(115.146.92.204)
Alive: 7 days
Type: m1.medium
Load: 6%

Cluster 2 - NeCTAR

CloudMan Console

Welcome to CloudMan. This application allows you to manage this instance cloud cluster and the services provided within. Your previous data store has been reconnected. Once the cluster has initialized, use the controls below to manage services provided by the application.

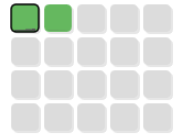| Terminate cluster | Add nodes ▾ | Remove nodes | Access Galaxy |

Status

Cluster name: cndr_ff
Disk status: 0 / 0 (0%)
Worker status: Idle: 0 Available: 0 Requested: 0
Service status: Applications ● Data ●

Master Node
i-0000556f
(115.146.92.146)
Alive: 7 days
Type: m1.medium
Load: 2%

Common resource pool

```
ubuntu@server-702d6c39-726a-42cb-901c-237f5fd86a19:~$ condor_q

-- Submitter: server-702d6c39-726a-42cb-901c-237f5fd86a19.novalocal : <115.146.92.204:96
 ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD

0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended
ubuntu@server-702d6c39-726a-42cb-901c-237f5fd86a19:~$ condor_status

Name              OpSys      Arch   State      Activity LoadAv Mem   ActvtyTime

slot1@server-702d6 LINUX      X86_64 Unclaimed Idle       0.000  3993  5+21:15:57
slot2@server-702d6 LINUX      X86_64 Unclaimed Idle       0.120  3993  5+21:15:37
slot1@server-c9851 LINUX      X86_64 Unclaimed Idle       0.000  4022  5+21:15:44
slot2@server-c9851 LINUX      X86_64 Unclaimed Idle       0.000  4022  5+21:15:45
slot3@server-c9851 LINUX      X86_64 Unclaimed Idle       0.120  4022  5+21:15:26
slot4@server-c9851 LINUX      X86_64 Unclaimed Idle       0.000  4022  5+21:15:47
slot5@server-c9851 LINUX      X86_64 Unclaimed Idle       0.000  4022  5+21:15:48
slot6@server-c9851 LINUX      X86_64 Unclaimed Idle       0.000  4022  5+21:15:49
slot7@server-c9851 LINUX      X86_64 Unclaimed Idle       0.000  4022  5+21:15:50
slot8@server-c9851 LINUX      X86_64 Unclaimed Idle       0.000  4022  5+21:15:43
                   Total Owner Claimed Unclaimed Matched Preempting Backfill

        X86_64/LINUX   10     0       0        10       0          0        0

             Total   10     0       0        10       0          0        0
```

Running jobs

```
condor_submit myjob.submit
condor_submit myjob.submit1
condor_submit myjob.submit2
condor_submit myjob.submit3
condor_submit myjob.submit4
condor_submit myjob.submit5
condor_submit myjob.submit6
condor_submit myjob.submit7
condor_submit myjob.submit8
condor_submit myjob.submit9
condor_submit myjob.submit10
condor_submit myjob.submit11
condor_submit myjob.submit12
condor_submit myjob.submit13
condor_submit myjob.submit14
```

Job submission script

```
executable=myprog
universe=vanilla
arguments=Example.$(Cluster).$(Process) 100
output=results.output.$(Process)
error=results.error.$(Process)
log=results.log.$x
notification=never
should_transfer_files=YES
when_to_transfer_output = ON_EXIT
queue
```

Conclusions

- Challenges
  - Data transfer & locality
- Future work
  - Streamline scaling of Condor hosts
  - Integration with Galaxy
  - Condor over Hadoop
- An architecture paper available from **MIPRO 2013**
  - "Support for data-intensive computing with CloudMan"

A cloud environment for distributed computing:
batch; Hadoop; HTCondor
http://usecloudman.org