A Secure Data Enclave and Analytics Platform
For Social Scientists

Yadu N. Babuji, Kyle Chard, Aaron Gerow, & Eamon Duede
Computation Institute, The University of Chicago and Argonne National Laboratory
{yadunand,chard,gerow,eduede}@uchicago.edu

2016 IEEE 12th Conference on eScience
Motivation

- Data driven research is ubiquitous. Data is fast becoming the defining assets for researchers, particularly those in the computational social sciences and humanities.

- Data is increasingly large; it is also valuable, proprietary, and sensitive.

- Social scientists (and other researchers) lack the technical and financial resources to securely and scalably manage large amounts of data while also supporting flexible and large-scale analytics.

- Cloud computing provides “infinite” storage and compute resources, however it requires technical expertise to deploy, configure, manage, and use.

- Cloud Kotta is a cloud-hosted environment that supports the secure management and analysis of large scientific datasets.
With private data-sets comes great responsibility

A significant fraction of the 10TB we manage is sensitive/proprietary data

**Web of Science** - from Thomson Reuters (1TB)

**UChicago AURA** grants DB - under NDA (~200GB)

**IEEE full texts** - under license (5.5TB)

We want to make this data accessible to our colleagues and collaborators, but secured within our infrastructure.
With massive data comes massive COST

We hold a tad over 10TB of research data.

10TB on EBS (SSD) = $1000 / mo
10TB on S3 (std) = $300 / mo
10TB on S3 (IA) = $125 / mo
10TB on Glacier = $70 / mo

Each comes with its own tradeoffs.
Large-scale data analytics

- Analyses are user driven and often interactive
- Development is often iterative
- Analyses are often compute intensive or memory intensive
- Complex analyses can be broken down to a many-task model (SPMD) and computed in parallel
- Scientific workloads are inherently sporadic and bursty (tracking submission deadlines)
- Variable lengths of time (minutes to weeks)
- Analyses are written in many languages (e.g., Python, Julia, BaSH, C++)
With massive compute comes massive COST

We’ve run over 75K* compute hours in 6 months.

On-demand = $15984.37
Spot-market (variable) = ~$4795.31
1 Reserved instance for 6mo = $17677.44

With i2.8xlarge, you can burn a 10K AWS credit in just 2 months.
We want to optimize for both cost and time-to-solution.

* Core hours
Solution
Cloud Kotta

- Cloud Kotta is a cloud-based platform that enables secure and cost-effective management and analysis of large, potentially sensitive data.
- The platform automatically provisions cloud infrastructure to host user submitted jobs.
- Data is migrated between storage tiers depending on access patterns and pre-defined policies.
- Role based access model for security.

* Pictured: Mehrangarh Fort at Jodhpur, Rajasthan
Automated storage management

Worker EC2

- EBS
  - $100/TB/mo

S3 Standard Bucket
- S3 Standard
  - $30/TB/mo

S3 Infrequent Access Bucket
- S3 Infrequent Access
  - $12.5/TB/mo

Glacier
- Glacier
  - $7/TB/mo
Elastic Provisioning
Security model

- Principle of least privilege throughout
- “Log in with Amazon”
- Users are assigned roles
- Policies permit access to resources for individual roles
- Instances are granted a trusted role that allows them to switch to a user role temporarily in order to inherit user permissions (e.g., access secure data)
- Compute layer is hosted within a private subnet enclosed within a VPC
Cloud Formation

- Security
- Auto Scaling
- Data Caching
User Interfaces

Web Interface

REST API

```
curl -i -H "Accept: application/json" http://52.2.217.165:8888/rest/v1/status
HTTP/1.1 200 OK
Content-Length: 3382
Content-Type: application/json
Date: Tue, 25 Oct 2016 05:42:38 GMT
Server: ip-10-0-0-03

{"status": "completed",
"items":
["0": {"username": "Yadu Nand B"},
"1": {"z_staging_dur": "0.01810562866197421975"},
"2": {"executable": "/bin/bash myscript.sh"},
"3": {"job_id": "d782d28d-44df-8aae-a2b6a4427f9"},
"4": {"complete_time": "2016-10-25 18:29:07"},
"5": {"submit_stamp": "2016-10-25 18:32:56"},
"6": {"z_processing_dur": "-0.098240261077888059375"},
"7": {"status": "completed"},
"8": {"outputs": "<a href="https://klab-j..."}
```
Data Interface

Upload Data

Browse Data
Job Submission
# Job management

## Job - Info

<table>
<thead>
<tr>
<th>Status</th>
<th>completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_published</td>
<td>1</td>
</tr>
<tr>
<td>z_stagen_dur</td>
<td>329.211119117495275390625</td>
</tr>
<tr>
<td>Executable</td>
<td>/bin/bash/myscript.sh</td>
</tr>
<tr>
<td>Job_id</td>
<td>08729b943-bebc-4135a5d3-8a9e9a2de818</td>
</tr>
<tr>
<td>Complete_time</td>
<td>2016-07-11 20:26:44</td>
</tr>
<tr>
<td>Publishdate</td>
<td>2016-07-13 16:19:22</td>
</tr>
<tr>
<td>Submit_stamp</td>
<td>2016-07-11 19:54:00</td>
</tr>
<tr>
<td>Processing_dur</td>
<td>1295.0844790936516357421875</td>
</tr>
<tr>
<td>Username</td>
<td>Wanqi Zhu</td>
</tr>
<tr>
<td>Inputs</td>
<td>article_influence.py</td>
</tr>
<tr>
<td>Description</td>
<td>Job Behavior: Computes the Article Influence of journals based on eigenfactor and article counts for # of papers published. Data obtained from web of science. Done in parallel insanely fast, but it's not necessary as it's not computationally intensive. Inputs: journal_counts_1995.tsv ~ journal_counts_2015.tsv, one per year, in tab-separated values of (including headers for the first row) JOURNAL COUNT eigenfactors_1995.tsv ~ eigenfactors_2015.tsv, one per year, in tab-separated values of (no headers) JOURNAL EIGENFACTOR Outputs: Calculate the Article Influence for each of the years and output to article_influence.csv. Also outputs intermediate values for debugging, but they are not useful.</td>
</tr>
<tr>
<td>Outputs</td>
<td>myscript.sh</td>
</tr>
<tr>
<td></td>
<td>article_influence.csv</td>
</tr>
<tr>
<td></td>
<td>t_counts.csv</td>
</tr>
<tr>
<td></td>
<td>j_counts.csv</td>
</tr>
<tr>
<td></td>
<td>j_ef.csv</td>
</tr>
<tr>
<td></td>
<td>STDERR.txt</td>
</tr>
<tr>
<td></td>
<td>STDOUT.txt</td>
</tr>
<tr>
<td>Start Time</td>
<td>2016-07-11 10:50:17</td>
</tr>
</tbody>
</table>

## CPU Utilization

![CPU Utilization Chart](chart1.png)

## Memory Utilization

![Memory Utilization Chart](chart2.png)
Early Usage/Results
System Utilization

Data Analyzed

vCPU time

Data (GB)


Days

vCPU hours

0 1000 2000 3000 4000 5000 6000 7000 8000

THE UNIVERSITY OF CHICAGO KNOWLEDGE LAB
Elastic scaling experiment

- To demonstrate the automatic scaling behavior we used a test-workload derived from historical production usage
- 40 jobs of 1, 3, or 4 hour durations with inter-arrival time from poisson-distribution ($\lambda = 0.1667$).
- Jobs simply call `sleep()`
- Each job uses a randomly selected data input of size $\{1,3,5,7,9\}$GB
- The scaling limit was set to a maximum of 40 nodes
- We plot the total nodes active and idle, as well as the state of each of the 40 jobs. X axis is time.
Early science on Cloud Kotta

- Text Analytics
- Matrix Factorization
- Optical Character Recognition (tesseract)
- Network Analysis
- Author-Topic models
Acknowledgements
Thanks

- Github repo: https://github.com/yadudoc/cloud_kotta
- Documentation: http://docs.cloudkotta.org/
- Support: yadunand@uchicago.edu