

# CS 550 Project

## An Empirical Evaluation of Distributed Key/Value Storage Systems

---

### Instructions:

- **Due date: 11:59PM on Sunday, 11/29/15**
- **Maximum Points: 100 points**
- **Extra Credit Points: 10 points**
- This is an individual assignment, although you may work in groups to brainstorm on possible solutions; your code implementation, report, and evaluation must be your own work.
- Please post your questions to the Piazza forum.
- Only a softcopy submission is required; it must be submitted to “Digital Drop Box” on Blackboard; please zip all files (report, source code, compilation scripts, and documentation) and submit it to BB; Name your file as this rule: “PROJ\_LASTNAME\_FIRSTNAME.{zip|tar|pdf}”. E.g. “Proj\_Raicu\_loan.tar”.
- **Late submission will be penalized at 10% per day (beyond the 7-day late pass).**

### 1 The problem

In this project, you are going to evaluate various distributed key/value storage systems. You must choose one of the two tracks outlined below for this assignment.

#### 1.1 Track 1: Amazon AWS

You are to use the Amazon AWS cloud to conduct your evaluation. You will have to [create an Amazon AWS account](#) (if you don’t already have one). Once you have an account, you can apply for an AWS Educational Credit at <http://aws.amazon.com/education/awseducate/apply/> (Under “Students”, click “Apply for AWS Educate for Students”). The educational credit should give you a \$35 credit that you can use towards the completion of this assignment, which should be sufficient if you are careful with your spending. You will conduct your evaluation using

Evaluate your system for both on the [Amazon AWS cloud](#) on up to 16 m3.medium instances (see for more details, <https://aws.amazon.com/ec2/instance-types/> and <https://aws.amazon.com/ec2/pricing/>). These on-demand instances cost \$0.067/hour, which should allow a single m3.medium instance to run for 522 hours, or with 16 instances at once, run for 32 hours. You can make your credits last longer by using spot instances (<https://aws.amazon.com/ec2/spot/>), which will cost you on average \$0.0094/hour (7X cheaper); this would allow you to run experiments at 16-node scales for over 9 days and still stay within the \$35 budget. The exact differences between on-demand and spot instances will be covered in class, but you should use spot instances whenever possible for this assignment.

You will use Amazon’s EC2 service (<https://aws.amazon.com/ec2/>) to provision your cluster of up to 16 instances. You could store your results in Amazon’s S3 service (<https://aws.amazon.com/s3/>), but you do not have to. You can use any Linux distribution for your OS.

#### 1.2 Track 2: Jarvis

You are to use the Jarvis Linux cluster to conduct your evaluation. You will have to create an account by visiting <https://bluesky.cs.iit.edu/jarvis>, and using userid “iit” and password “iit2014”. This userid and password is only to request an account. More information about the Jarvis cluster can be found at <http://www.cs.iit.edu/~iraicu/teaching/CS550-F15/intro-jarvis.pdf>. Jarvis has 20 nodes, and runs a job manager that you must use to share this cluster with your classmates. There is no virtualization, and each node has 6-cores and 8GB or 16GB of memory; each node has some local disk, 1Gb/sec network, and a shared file system NFS. These nodes are likely more powerful than the Amazon m3.medium instances, so a

direct comparison between the Jarvis cluster and the Amazon AWS will not be possible; the more similar instance types on AWS are m3.xlarge or m3.2xlarge.

### 1.3 Systems to Evaluate

You must choose 3 of the following 10 distributed key/value stores to evaluate:

- Amazon DynamoDB (<https://aws.amazon.com/dynamodb/>)
- MongoDB (<https://www.mongodb.org>)
- Cassandra (<http://cassandra.apache.org>)
- CouchDB (<http://couchdb.apache.org>)
- HBase (<http://hbase.apache.org>)
- Riak (<http://basho.com/products/#riak>)
- ZHT (<http://datasys.cs.iit.edu/projects/ZHT/>)
- Redis (<http://redis.io>)
- HyperTable (<http://www.hypertable.com>)
- Oracle NoSQL Database (<http://www.oracle.com/technetwork/database/database-technologies/nosqldb/overview/index.html>)

And compare to your system you implemented for your Programming Assignment #2; give your system a name (other than ZHT) for easier discussion and presentation of your results.

### 1.3 Evaluation Scale and Metrics

Read this paper on ZHT ([http://datasys.cs.iit.edu/publications/2015\\_CCPE-zht.pdf](http://datasys.cs.iit.edu/publications/2015_CCPE-zht.pdf)). Particularly, Figure 6, Figure 7, Figure 9, Figure 11, Figure 12, Figure 13, and Figure 14 are very important. You will have to conduct enough experiments to generate 8 figures, 4 for latency and 4 for throughput; the 4 figures comes from presenting each operation insert/lookup/remove separately, and as an average across all 3 operations.

On each instance/node, a client-server pair is deployed. Test workload is a set of key-value pairs where the key is 10 bytes and value is 90 bytes. Clients sequentially send all of the key-value pairs through a client API for insert, then lookup, and then remove. Your keys should be randomly generated, which will produce an All-to-All communication pattern, with the same number of servers and clients.

The metrics you will measure and report are:

- **Latency:** Latency presents the time per operation (insert/lookup/remove) taken from a request to be submitted from a client to a response to be received by the client, measured in milliseconds (ms). Note that the latency consists of round trip network communication, system processing, and storage access time
- **Throughput:** The number of operations (insert/lookup/remove) the system can handle over some period of time, measured in Kilo Ops/s

Make the necessary plots to visualize your data. Explain why your results make sense; what explicit things did you do to verify that your performance as you expected?

### 1.4 Poster Presentation

You must present your project results in a poster format, 2 feet by 3 feet large. An example of a poster can be found at [http://datasys.cs.iit.edu/reports/2013\\_GCASR13\\_poster102.pdf](http://datasys.cs.iit.edu/reports/2013_GCASR13_poster102.pdf), but many more posters can be found at <http://datasys.cs.iit.edu/reports/index.html>. Posters should be self contained, have a title, author list, abstract, motivation, proposed work, evaluation, conclusions, and references. You should not use paragraph style writing, except for the abstract perhaps; the poster should be primarily be composed of

figures and bullet points. You will be giving a live oral presentation of your poster (<3 minutes) and participate in a Q&A session on November 30<sup>th</sup> or December 2<sup>nd</sup> during class. For online students, you must record a video of up to 3-minutes long presenting your poster, and submit it via BB (note that large video files will not be accepted, and they must be hosted elsewhere, and only the URL should be submitted).

You must print your poster and bring it with you to class on November 30<sup>th</sup> and December 2<sup>nd</sup> (if you don't present on November 30<sup>th</sup>). Absence from these live oral presentations will result in a 0 for the oral presentation score. There will be no makeup oral presentation.

## 2 What you will submit & Grading

When you have finished implementing the complete assignment as described above, you should submit your solution to 'digital drop box' on blackboard. Each program must work correctly and be detailed in-line documented. You should hand in:

1. **Source Code (20 points):** You must hand in all your source code of any scripts you used to automate your performance evaluation. If you wrote any programs specific to each system in order to perform the evaluation, include these evaluation drivers.
2. **Poster (40 points):** You must create a 2 foot by 3 foot poster to present your evaluation; you will be scored on completeness, correctness, organization, and visual appeal.
3. **Oral Presentation (40 points):** You will have to present your project through a live Q&A session on November 30<sup>th</sup> or December 2<sup>nd</sup> during class; for online students, you must record a video of up to 3-minutes long presenting your poster, and submit it via BB.
4. **Extra Credit (10 points):** Add a 4<sup>th</sup> system to the evaluation

Please put all of the above into one .zip or .tar file, and upload it to 'digital drop box' on blackboard'. The name of .zip or .tar should follow this format: "PROG#\_LASTNAME\_FIRSTNAME.{zip|tar|pdf}".

Please do NOT email your files to the professor and TA!!

Grades for late programs will be lowered 10% per day points per day late (beyond the 7-day late pass).