CSCI:6502
Group Project

Streaming Log Analytics Tool

Team Members:
Janani Selvan (jase6406): 001
Nithin Veer Reddy (nika9944): 001
Shruthi Sridharan (shsr7296): 001B
Veena Prasad (vepr4844): 001

Project Motivation

- Logs are generated across multiple platforms - mostly in raw format
- Although the successful logs does not generate any insightful information but error logs should be analysed properly in order to have a successful flow.
- The information from these logs should be extracted and then further used to rectify the system.
- As most of the systems do have its own style of log formats so each log generator should have a different pipeline to merge logs from multiple sources
- There is an essential need to merge the logs from multiple sources to extract some meaningful information.

Related Work

- LogDriver - analyzes the application-level resiliency in extreme-scale computing systems.
  - Capable of handling data generated by system monitoring tools in Blue Waters and scalable tool implemented in mapreduce frameworks.
  - Answers the research question about the complexity involved in automatic parsing of heterogeneous log messages in seconds as analysis tools prefer the logs in a unified format.
  - Applications - security, anomalies detection, software system maintenance.

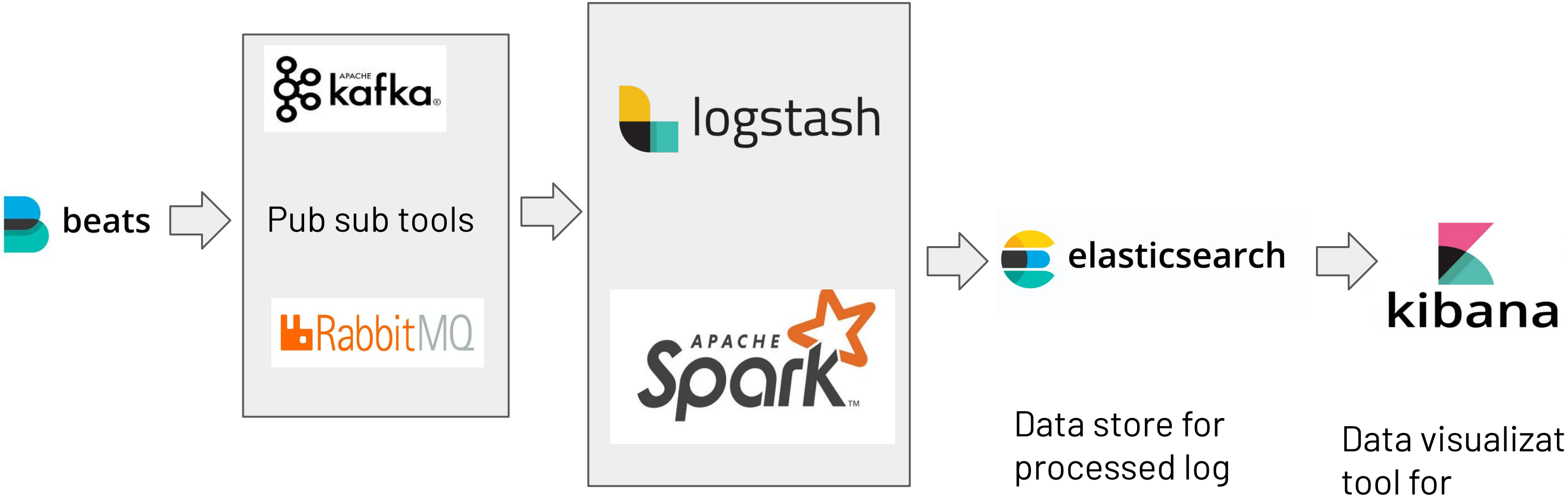
Our techniques

- Ingest the standalone logs into two streaming pipelines according to the log categories.
- Analyzing the performance of the log parser using the ElasticSearch and Spark.
- Visualize the processed logs to understand any errors, irregularities thereby giving some valuable insights to rectify the same.
- Visualizations are prepared on kibana which includes
  - Dashboards,
  - Graphs, Charts, Word Clouds
  - Dailly, Weekly (Regular timely) reports

Dataset

- Data is sourced from the publicly available repositories.
- Most of the data is being sourced from a single repository logpai/loghub
- Overall data is close to 77GB
- Some of the popular system logs includes data from the following technologies
    - Distributed System Logs
    - Operating System Logs
    - Supercomputer Logs
    - Server Application Logs
    - Mobile Application Logs

Design choices



Pub sub tools

Log processing tools

Data store for processed log data

Data visualization tool for Elasticsearch

Reasoning of our choices

Tool	**Purpose**
Elasticsearch	Stores and queries the data ingested
CloudLab	Hosting the data, es clusters, applications
Kibana	Build the visualizations and dashboards
Logstash	Pipeline to push the data from sources to ES
Spark	To map/reduce the data into meaning aggregations

The Work

- End to end completion of Major log groups - Android, Spark , Zookeeper, Hadoop logs.
- Complete pipeline is established with relevant infrastructure has been deployed.
- Constructed both Spark & Elasticsearch pipeline and thereby compared the performance.
- Out of the various logs available, we analysed and created clusters and processed one major logs pattern from each cluster of logs.
- Logs are further analysed according to the log format before parsing them to respective pipelines.
- Around 700K logs are parsed so far.
- An End to End Framework built which would be cross utilized to parse other logs in the same cluster.
- Some meaningful analysis are projected on kibana dashboards.

ElasticSearch - Accomplishment

- Constructed parsing scripts for each and every log category and thereby indexing the same in the elasticsearch.

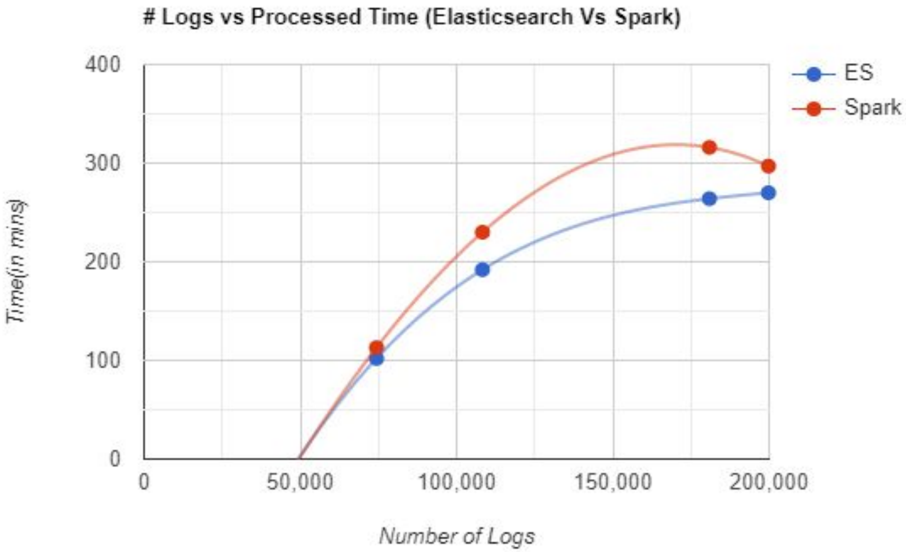
Log type	# logs	Log file size (MB)	Processing time (in minutes)
Android	199799	192	270
Hadoop	180897	5.1	264
Zookeeper	74380	10	102
Spark	108291	2950	192

Spark - Accomplishment

- Implementation of alternate pipeline - Spark based on the log categorization and thereby ingesting the standalone logs into the Elasticsearch.

Log type	# logs	Log file size(MB)	Processing time(in minutes)
Android	199799	192	297
Hadoop	180897	5.1	316
Zookeeper	74380	10	113
Spark	108291	2950	230

Graphical Comparison



Key Observations

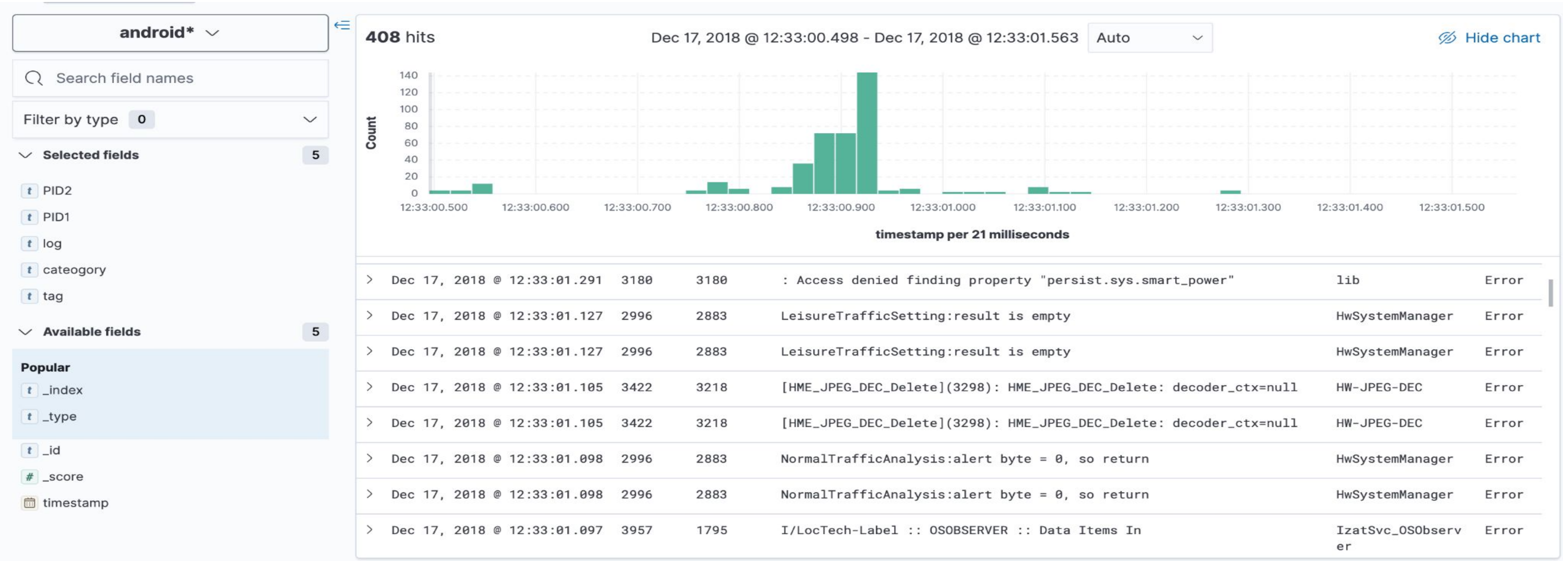
- End to end completion of Major log groups - Android, Spark , Zookeeper, Hadoop logs.
- Complete pipeline is established with relevant infrastructure has been deployed.
- Constructed both Spark & Elasticsearch pipeline and thereby compared the performance.
- Now that the load wasn't distributed, the simpler pipeline with Elastic search performed better.
- Cost benefit analysis with distributed load across transient servers might help us indicate the most cost-effective pipeline.

Kibana Dashboard links

LOGS	KIBANA LINK	DATE TIME FRAME
Android	http://ms1133.utah.cloudlab.us:6100/goto/2108e0410c3ed33844553123c525d133	Dec 17, 2018 00:00:00 - Dec 18, 2018 00:00:00
Hadoop	http://ms1133.utah.cloudlab.us:6100/goto/7cc69df5706e21f216241723fe1081de	Oct 18, 2015 11:00:00 - Oct 19, 2015 12:00:00
Spark	http://ms1133.utah.cloudlab.us:6100/goto/3a0c2c0cf38254a967a441789aa616b0	Jun 17, 2009 00:00:00 - Jun 17, 2009 20:30:00
Zookeeper	http://ms1133.utah.cloudlab.us:6100/goto/9a6f6dfbccb3411b40dd260fd9497c75	Jul 28, 2015 00:00:00 - Aug 29, 2015 00:00:00

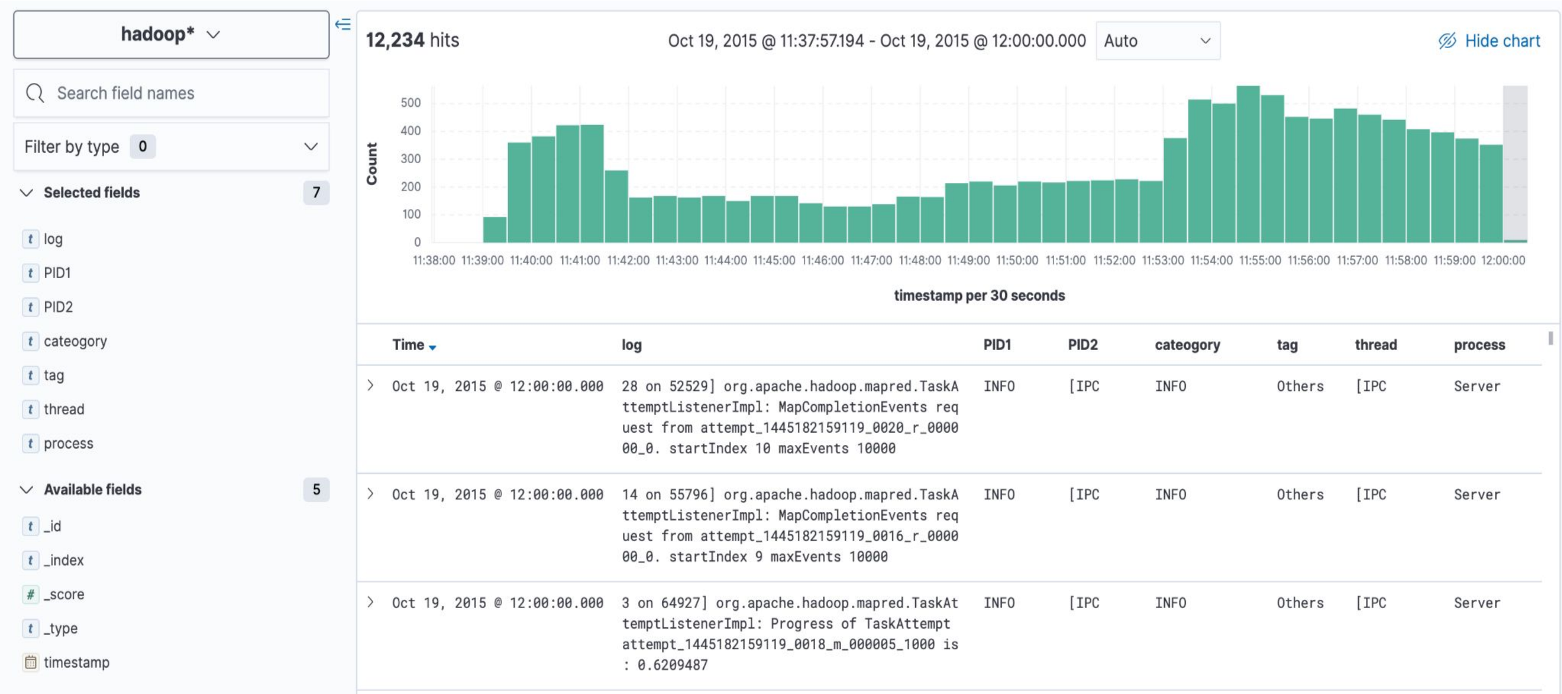
Insights into the log data

Visualizations are prepared on Kibana dashboard to provide valuable insights to rectify and help in software production.



```
>  Dec 17, 2018 @ 12:33:01.291  3180   3180   : Access denied finding property "persist.sys.smart_power"       lib              Error
>  Dec 17, 2018 @ 12:33:01.127  2996   2883   LeisureTrafficSetting:result is empty                            HwSystemManager  Error
>  Dec 17, 2018 @ 12:33:01.127  2996   2883   LeisureTrafficSetting:result is empty                            HwSystemManager  Error
>  Dec 17, 2018 @ 12:33:01.105  3422   3218   [HME_JPEG_DEC_Delete](3298): HME_JPEG_DEC_Delete: decoder_ctx=null  HW-JPEG-DEC      Error
>  Dec 17, 2018 @ 12:33:01.105  3422   3218   [HME_JPEG_DEC_Delete](3298): HME_JPEG_DEC_Delete: decoder_ctx=null  HW-JPEG-DEC      Error
>  Dec 17, 2018 @ 12:33:01.098  2996   2883   NormalTrafficAnalysis:alert byte = 0, so return                  HwSystemManager  Error
>  Dec 17, 2018 @ 12:33:01.098  2996   2883   NormalTrafficAnalysis:alert byte = 0, so return                  HwSystemManager  Error
>  Dec 17, 2018 @ 12:33:01.097  3957   1795   I/LocTech-Label :: OSOBSERVER :: Data Items In                   IzatSvc_OSObserv Error
                                                                                                                er
```

Insights into the log data

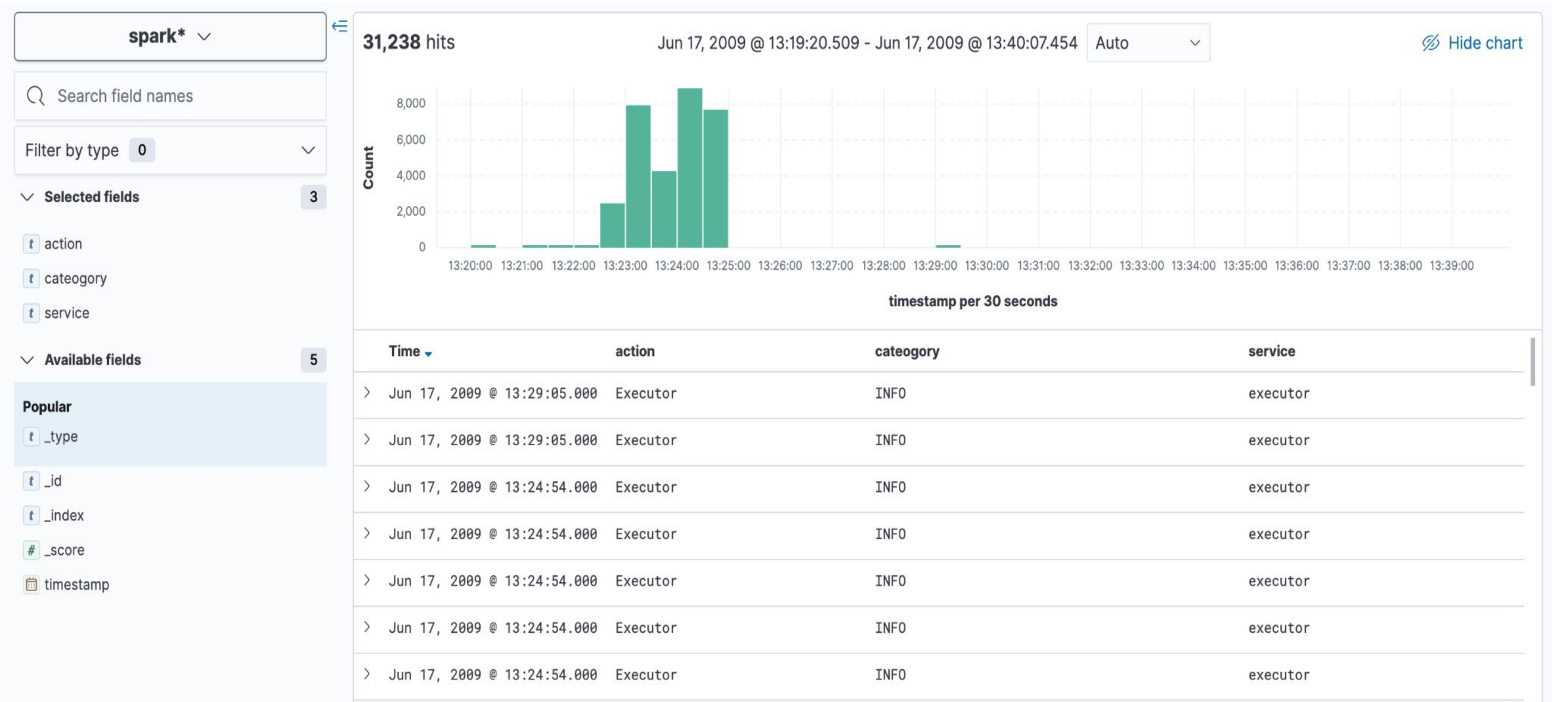


12,234 hits

Oct 19, 2015 @ 11:37:57.194 - Oct 19, 2015 @ 12:00:00.000

Time	log	PID1	PID2	cateogory	tag	thread	process
Oct 19, 2015 @ 12:00:00.000	28 on 52529] org.apache.hadoop.mapred.TaskAttemptListenerImpl: MapCompletionEvents request from attempt_1445182159119_0020_r_000000_0. startIndex 10 maxEvents 10000	INFO	[IPC	INFO	Others	[IPC	Server
Oct 19, 2015 @ 12:00:00.000	14 on 55796] org.apache.hadoop.mapred.TaskAttemptListenerImpl: MapCompletionEvents request from attempt_1445182159119_0016_r_000000_0. startIndex 9 maxEvents 10000	INFO	[IPC	INFO	Others	[IPC	Server
Oct 19, 2015 @ 12:00:00.000	3 on 64927] org.apache.hadoop.mapred.TaskAttemptListenerImpl: Progress of TaskAttempt attempt_1445182159119_0018_m_000005_1000 is : 0.6209487	INFO	[IPC	INFO	Others	[IPC	Server

Insights into the log data



31,238 hits

Jun 17, 2009 @ 13:19:20.509 - Jun 17, 2009 @ 13:40:07.454

Selected fields 3

- t action
- t cateogory
- t service

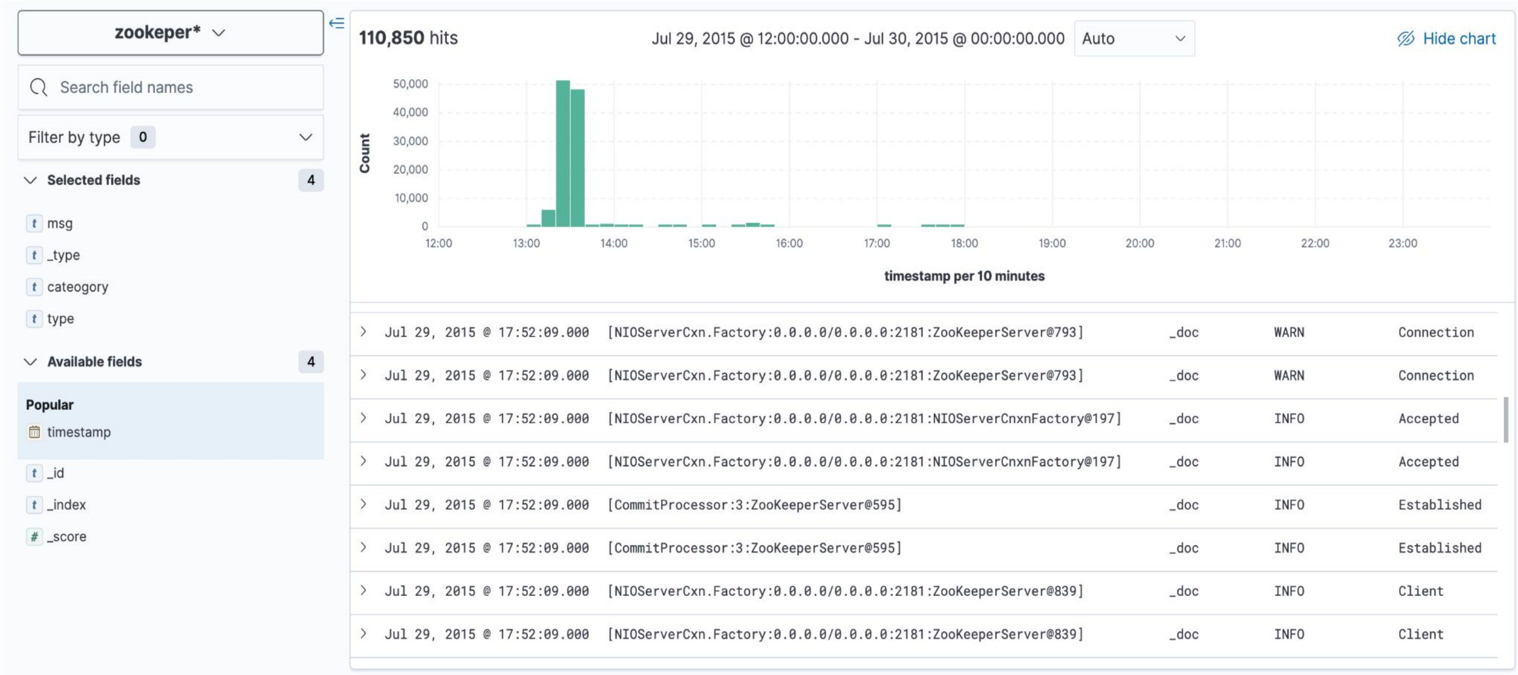
Available fields 5

Popular

- t _type
- t _id
- t _index
- # _score
- timestamp

Time	action	cateogory	service
Jun 17, 2009 @ 13:29:05.000	Executor	INFO	executor
Jun 17, 2009 @ 13:29:05.000	Executor	INFO	executor
Jun 17, 2009 @ 13:24:54.000	Executor	INFO	executor
Jun 17, 2009 @ 13:24:54.000	Executor	INFO	executor
Jun 17, 2009 @ 13:24:54.000	Executor	INFO	executor
Jun 17, 2009 @ 13:24:54.000	Executor	INFO	executor
Jun 17, 2009 @ 13:24:54.000	Executor	INFO	executor

Insights into the log data



110,850 hits

Jul 29, 2015 @ 12:00:00.000 - Jul 30, 2015 @ 00:00:00.000

```
Jul 29, 2015 @ 17:52:09.000  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:2181:ZooKeeperServer@793]        _doc  WARN  Connection
Jul 29, 2015 @ 17:52:09.000  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:2181:ZooKeeperServer@793]        _doc  WARN  Connection
Jul 29, 2015 @ 17:52:09.000  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:2181:NIOServerCnxnFactory@197]   _doc  INFO  Accepted
Jul 29, 2015 @ 17:52:09.000  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:2181:NIOServerCnxnFactory@197]   _doc  INFO  Accepted
Jul 29, 2015 @ 17:52:09.000  [CommitProcessor:3:ZooKeeperServer@595]                                _doc  INFO  Established
Jul 29, 2015 @ 17:52:09.000  [CommitProcessor:3:ZooKeeperServer@595]                                _doc  INFO  Established
Jul 29, 2015 @ 17:52:09.000  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:2181:ZooKeeperServer@839]        _doc  INFO  Client
Jul 29, 2015 @ 17:52:09.000  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:2181:ZooKeeperServer@839]        _doc  INFO  Client
```

Evaluation

- Unlike routine way of evaluation using some metrics, here we put evaluations in terms of analyzing and comparing.
- For each set of log components, we Analyze and Compare the performance of logs
- Results are evaluated and ranked according to the tools of each component alongside with the power/time/cost and nature of the logs.
- Measure of success:
  - Spark - Process almost 200k line items in 300 minutes
  - ELK - Process almost 200k line items in 270 minutes

Conclusion

- Set up experiments to analyze performance of ElasticSearch and Spark based pipeline based on 3 V's (Volume, Velocity and Variety)
- Comparing the processing times of ElasticSearch and Spark, it can be seen that ElasticSearch has lesser processing time.
- On qualitative aspects of growing the application and bringing a scalable solution, Spark would work better.
- Some insightful information about the logs were generated using Kibana.
- An End to End Framework built which would be cross utilized to parse other logs in the same cluster.

Future work

- As we have parsed most of the logs which are static, we could extend the same work for a live stream of logs.
- We could have modified the existing pipeline to honour the log streams.
- Although there won't be much difference between Batch Processing Vs Live Processing, the results would have been interesting to watch.
- Distribute tasks across multiple servers and do a CBA across different pipelines.

References

- https://github.com/logpai/loghub
- https://www.elastic.co/
- https://www.elastic.co/beats/
- https://www.elastic.co/guide/en/elasticsearch/hadoop/master/spark.html
- https://ci.apache.org/projects/flink/flink-docs-stable/dev/connectors/elasticsearch.html
- https://www.elastic.co/guide/en/logstash/current/plugins-inputs-kafka.html
- https://www.elastic.co/guide/en/logstash/current/plugins-inputs-rabbitmq.html
- https://www.elastic.co/blog/elastic-stack-primer
- https://www.elastic.co/guide/en/elasticsearch/reference/master/data-streams.html
- https://logpai.github.io