Using ML to Automate Dynamic Error Categorization

Antonio Davoli
Production Engineer, Servers Lifecycle Engineering
Agenda

• Servers Lifecycle
• Clustering
• SQClusters
• Results and future work
Servers Lifecycle
Servers Lifecycle

Distributed Jobs Orchestrator for handling server lifecycle stages (e.g. Provisioning)

- Hardware Validation
- Firmware checks and updates
- OS installation and basic packages
- Asset Custom Configurations
Suspended Jobs Queue be like:

Image Credit: https://xkcd.com/1106
“if you torture the data long enough, it will confess”

Ronald Coase, Economist
Moar data!

- Logs
  - Error Code and Message
  - Error Events
- Operating System
  - Major Version
  - Minor Version
- Location
  - Datacenter
  - Rack
- Firmware
  - Network
  - Motherboard
Inferring Similarities

Considered all the various data sources we can pull data from, why don’t we try to infer more similarities that we can exploit to fix the highest number of servers in the shorter possible time?
Clustering
Clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups.

— Wikipedia
Clustering Algorithms

- Centroids Models
  - K-means
  - K-means++

- Distribution Models
  - Gaussian Mixtures

- Density Models
  - Optics
  - DBSCAN

- Graph
  - HCS

- Neural Models
  - SOFM
SQClusters
SQClusters

Applying DBSCAN to the Orchestrator Suspend Queue

DBSCAN is a density-based clustering algorithm.

Given a set of points in some space, it groups points that are closely packed together, marking as outliers points that lie alone in low-density regions.

DBSCAN
Algorithm Internals

*Doesn’t require to specify the number of clusters*, it does have a notion of noise which makes it robust to outliers.

- $\varepsilon$ (eps): minimum distance between points in space,
- **min_points**: minimum number of points required to form a dense region
K-means  dbscan

One-Hot Encoding for Categorical Features

Categorical features are substituted by their integer representation.

<table>
<thead>
<tr>
<th>Server</th>
<th>Datacenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Singapore</td>
</tr>
<tr>
<td>2</td>
<td>Sweden</td>
</tr>
<tr>
<td>3</td>
<td>Ireland</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Server</th>
<th>Datacenter_Singapore</th>
<th>Datacenter_Sweden</th>
<th>Datacenter_Ireland</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Hash values for clusters identifiers
SQClusters Pipeline

1. Retrieve list of suspended jobs
2. Extract servers augmented information
3. Fuse data and execute clustering algorithm
4. Upload found clusters in results dataset
Real example of clustering results

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Size</th>
<th>Error Message</th>
<th>Hostname Scheme</th>
<th>Model</th>
<th>Datacenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>abc</td>
<td>231</td>
<td>chef_error_msg</td>
<td>hadoop</td>
<td>Model #1</td>
<td>SGP, SWE</td>
</tr>
<tr>
<td>xyz</td>
<td>91</td>
<td>dhcp_error_msg, pxe_boot_error_msg</td>
<td>cache</td>
<td>Model #2</td>
<td>IRL</td>
</tr>
</tbody>
</table>
Lessons learned

- Structured logging helps (use it, it’ll pay back!),
- Spend all the time you need in cleaning your data,
- When you do this sort of exploratory work, listen to your data and make them “confess”,
- Using ML tooling is extremely easy to use: `dbscan.fit(X)`
What next?

- Experiments with more clustering algorithms, especially hierarchical approach based on density,
- Improve hashing techniques,
- Extract data on trends analysis and seasonality
Questions?
Backup: DBSCAN Internals

- NearestNeighbors based (Pair-wise or KD-Tree)
- Depth-first search, very similar to the classic algorithm for computing connected
Backup: k-means Internals

• Iterative approach (Expectation–Maximization), continues to compute centroids continuously
• The "cluster center" is the arithmetic mean of all the points belonging to the cluster.