BeRLOS

Solr: Setup and Benchmark

Yatindra Indoria, 20CS30060
Chirag Ghosh, 20CS10020
Srishty Gandhi, 20CS30052
Shreya Agarwal, 20CS10058
Sarita Singh, 20CS10053

Deliverables

1. Setup scripts, and parameter tuning utility for initial setup of a distributed search application in Apache Solr.
2. A benchmark, as well as a utility to do the benchmarking (possibly might outsource).
3. A GUI to manage the deployment, and monitor the current statistics for management.

Setup Scripts and Parameter Tuning Utility

Overview:
The goal of this part is to provide a set of setup scripts and a parameter tuning utility that can be used to set up and configure a distributed search application in Apache Solr. The setup scripts should automate the process of installing and configuring Solr on a cluster of machines, while the parameter tuning utility should allow users to easily adjust the various configuration parameters to optimize performance.

Functional Requirements:

- The setup scripts should be able to automate the installation of Apache Solr on a cluster of machines. The script should be able to detect/ input the number of nodes in the cluster and install Solr accordingly.
- The setup scripts should be able to configure the Solr cluster based on the user's requirements, such as replication factor, sharding, etc.
- The parameter tuning utility should allow users to adjust various configuration parameters, such as cache sizes, JVM heap size, and Solr indexing parameters.
- The parameter tuning utility should provide guidance on how to optimize the parameters based on the Solr deployment.

Non-functional Requirements:

- The setup scripts should be horizontally scalable (able to handle large clusters).
- The setup scripts should be platform-agnostic and work on all major operating systems*.
- The parameter tuning utility should be user-friendly and have a simple command-line interface.
- The parameter tuning utility should be able to provide recommendations for optimal configuration settings based on the deployment.

* will be done depending upon the availability of the time

Benchmarking Utility

Overview:
The goal of this part is to provide a benchmarking utility that can be used to test the performance of a high QPS Apache Solr server. The benchmarking utility should be able to generate high QPS traffic and measure the response time of the Solr server.

Functional Requirements:

- The benchmarking utility should be able to generate a configurable number of concurrent requests to the Solr server.
- The benchmarking utility should be able to measure the response time of the Solr server for each request and calculate the average response time.
- The benchmarking utility should be able to generate realistic queries that represent typical usage patterns for the Solr deployment.

Non-functional Requirements:

- The benchmarking utility should be able to generate traffic with a QPS of at least 10,000 per node.
- The benchmarking utility should be platform-agnostic and work on all major operating systems*.
- The benchmarking utility should be horizontally scalable on Solr deployments.
- The benchmarking utility should be able to provide detailed reports on the performance of the Solr server.

* will be done depending upon the availability of the time

GUI for Deployment and Monitoring

Overview:
The goal of this part is to provide a user-friendly GUI that can be used to deploy and monitor the Solr deployment. The GUI should be able to provide real-time statistics on the performance of the Solr server.

Functional Requirements:

- The GUI should provide real-time statistics on the performance of the Solr deployment, including QPS, response time, and server utilization.
- The GUI should provide alerts and notifications when performance metrics fall below acceptable levels.
- The GUI should provide a dashboard that shows the overall health of the Solr deployment.

Non-functional Requirements:

- The GUI should be user-friendly and easy to navigate.
- The GUI should be web-based and work on all major browsers.
- The GUI should be customizable and allow users to configure which statistics are displayed on the dashboard.

-*-

* will be done depending upon the availability of the time