پروژه

شرح پروژه

هدف اصلی این پروژه مقایسه بعضی از تکنولوژی‌های موجود در دنیای ذخیره و بازیابی اطلاعات است. این پروژه به دو قسمت برای OLAP و OLTP تقسیم شده است. طبیعتا نیاز متفاوت این دو حوزه، تکنولوژی های متفاوتی را می طلبد.

قسمت OLAP

برای این قسمت از بنچ مارک استاندارد TPC-H ورژن 2.17.1 استفاده میکنیم که مورد تایید صنعت و دانشگاه است. درباره این استاندارد منبع [1] را مطالعه کنید. حدود 100 گیگ دیتا روی سرورها در اختیار شما قرار گرفته است (فلدر /data/OLAP_Benchmark_data) که در قالب جداول استاندارد TPC-H می‌باشد. شکل صفحه 13 منبع [1] روابط این جداول را نشان میدهد.

تمام کوئری‌هایی که در این قسمت اجرا می‌کنید کویرهای استاندارد این پلتفرم هستند و روی دیتایی که در اختیار شما قرار گرفته اجرا می‌شوند (اجازه ندارید دیتای دیگری را استفاده کنید). تعداد کویری ها 22 تا است و کد SQL آنها در فلدر (/data/OLAP_Benchmark_queries) روی سرور قرار دارد. در این قسمت از شما میخواهیم که این 22 کویری را با استفاده از تکنولوژی های زیر روی دیتا اجرا کنید و گزارش اجرا را طبق فرمت مشخص شده در قسمت گزارش تحویل دهید.

تکنولوژی های مورد استفاده در قسمت OLAP:

- PostgreSQL

دیتای داده شده را در داخل یک دیتابیس PostgreSQL لود کنید و کویری ها را با زبان اس کیو ال اجرا کنید. برای ساده شدن کار می‌توانید جداول را با استفاده کویری هایی که در منبع [5] داده شده‌اند بسازید. اگر نیاز دارید در مورد PostgreSQL بیشتر بدانید به [8] مراجعه کنید.

- Spark + Parquet + HDFS

دیتا را در قالب فرمت Parquet دربیارین و روی HDFS قرار دهید سپس با استفاده از اسپارک 22 کویری داده شده را پیاده سازی و اجرا کنید. اگر نیاز دارید در مورد Spark بیشتر بدانید به [9] مراجعه کنید. اگر نیاز دارید

در مورد HDFS بیشتر بدانید به [10] مراجعه کنید. در مورد ترکیب Parquet با Spark نیز این منبع را ببینید [11].

- Spark + ORC + HDFS

مانند قسمت قبل است فقط دیتا را به قالب ORC دربیارید [12].

- Spark + Avro + HDFS

مانند قسمت قبل است فقط دیتا را به قالب Avro دربیارید [13].

- Flink + Avro + HDFS

مانند قسمت قبل است فقط باید پیاده سازی کویری ها در اسپارک را به پیاده سازی روی Flink تغییر دهید [14].

قسمت OLTP

در این قسمت هدف مقایسه کارایی دو دیتابیس NewSQL با PostgreSQL است. از اونجایی که تولید بار OLTP نیاز به پیاده‌سازی منطق تراکنش‌های پیچیده دارد از پلتفرم‌های آماده استفاده خواهیم کرد. در این قسمت بر مبنای TPC-C عمل خواهیم کرد که استانداردی مقبول در صنعت و دانشگاه است [2].

برای این قسمت پروژه، یکی از دوتایی‌های زیر را انتخاب کنید و کارایی آنها در زیر بار OLTP مقایسه کنید (هدف مقایسه کارایی VoltDB یا CockroachDB با PostgreSQL است).

- PostgreSQL and VoltDB [15]
- PostgreSQL and CockroachDB [16]

برای تولید بار OLTP از پلتفرم OLTPBench استفاده کنید [3]. برای استفاده از این پلتفرم نیاز دارید که از jdbc درایور دیتابیس‌های تحت تست بهره ببرید. حتما مطمین شوید که از این پلتفرم در قالب TPC-C استفاده کنید. برای شروع کار با این پلتفرم مستندات [4] نقطه شروع خوبی است. همانطور که در [4] توضیح داده شده این برنامه از jdbc درایور استفاده میکند و کافی است کلاس بنچ‌مارک را برابر مقدار tpcc ست کنید.

خروجی پروژه

در انتهای پروژه یک گزارش به ازای هر کدام از قسمت‌های OLAP و OLTP بنویسید. در هر کدام از گزارش‌ها مشخصات سیستمی و محیطی اجرای تست را کامل ذکر کنید. پارامترهای شخصی‌سازی شده در سیستم مورد تست را نیز بیان کنید. در قسمت OLAP انتظار این است که گزارش در مورد کارایی سیستم از لحاظ سرعت و حافظه به ازای هر یک از کویری‌ها گزارش شود (نمونه یک گزارش را در [6] ببینید). در انتهای گزارش OLAP همه پیاده‌سازی‌های کویری‌ها را به صورت پیوست ارایه دهید. همچنین اگر برای تبدیل فرمت اسکریپتی را توسعه داده‌اید آن را نیز در گزارش بیاورید. تمام کدهای پیاده سازی را باید روی گیت قرار دهید و لینک آن را در گزارش ذکر نمایید (همه کد، حتی اسکریپت های تبدیل فرمت).

در قسمت OLTP خروجی کار شما وابسته به پلتفرم OLTPBench است که خروجی این پلتفرم در [7] توضیح داده شده است. در گزارش OLTP انتظار میرود که کارایی سیستم مورد تست را در معیارهای مختلف به ازای پارامترهای مختلف پلتفرم بدست آورده و گزارش کنید. برای این قسمت می‌توانید از trace analyzer خود پلتفرم بهره ببرید [7].

بدیهی است هرگونه کد کپی شده از اینترنت یا دوستانتان مصداق تقلب است.

لیست دانشجویان و سرورهای اختصاص داده شده جهت انجام پروژه در لینک زیر قرار داده شده است. شما میتوانید از این سرورها فقط و فقط برای انجام پروژه این درس استفاده کنید و هرگونه استفاده دیگر از این سرورها کل کلاس را از داشتن منابع محاسباتی محروم میکند. دقت کنید بار این سرورها تمام مانیتور می‌شوند. برای دسترسی به این سرورها به owner آن سرور که در فایل زیر مشخص شده مراجعه کنید و کلید ssh خود را به وی برسانید.

https://docs.google.com/spreadsheets/d/14XWdKJZYS1d5nnZHCA3_x-IJUclo5OGuBWicQOmNvUk/edit#gid=952294389

تکنولوژی‌های زیر به صورت داکر روی سرور در اختیار شما قرار میگیرد:

- Spark
- HDFS
- PostgreSQL
- Flink
- VoltDB
- CockroachDB

[1] http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-h_v2.17.1.pdf

[2] http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-c_v5.11.0.pdf

[3] https://github.com/oltpbenchmark/oltpbench

[4] https://github.com/oltpbenchmark/oltpbench/wiki/Quickstart

[5] https://github.com/dragansah/tpch-dbgen/blob/master/tpch-create.sql

[6] https://courses.cs.washington.edu/courses/cse544/13sp/final-projects/p18-lijl.pdf

[7] https://exascale.info/assets/pdf/p260-difallah.pdf

[8] https://www.postgresql.org/docs/9.4/static/tutorial-start.html

[9] https://spark.apache.org/docs/latest/quick-start.html

[10] https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html

[11] https://databricks.com/session/spark-parquet-in-depth

[12] https://hortonworks.com/blog/bringing-orc-support-into-apache-spark/

[13] https://docs.databricks.com/spark/latest/data-sources/read-avro.html

[14] https://ci.apache.org/projects/flink/flink-docs-stable/dev/batch/connectors.html#avro-support-in-flink

[15] https://github.com/voltdb/voltdb

[16] https://www.cockroachlabs.com/docs/stable/build-an-app-with-cockroachdb.html