Tuning and Optimizing an end to end benchmark

Our Experience tuning Big Data Benchmark for BigBench Specification

Yi Zhou
Intel SSG/STO/Big Data Technology
Contact us: sto-bigdata-qa-prc@intel.com
About us

- Intel SSG STO big data performance tuning team
- Working on performance tuning and optimization for Hadoop ecosystem since 2011
- Contributors of Big-Bench, collaboration with the Apache Open Source community to improve the performance with Big-Bench benchmark
Today’s Agenda

- Why Big-Bench
- Contribution to Big-Bench
- Big-Bench Query Optimization
- Tuning Spark SQL with Big-Bench
Why Big-Bench

- End-to-End benchmark v.s. Micro-Benchmark
- Multiple popular big data processing engines
- Representative workload
- Open Source standard based
- Industry consortium proposed
- Support from 10+ ecosystem partners
Contribution to Big-Bench

- Extend the mainstream Hadoop engines: Spark SQL, Hive on Spark, Impala...
- Query optimization
- Query level engine setting
Example: Query 10

```sql
... 
INSERT INTO TABLE ${hiveconf:RESULT_TABLE}
SELECT extract_sentiment(pr_item_sk,pr_review_content) AS (pr_item_sk, review_sentence, sentiment, sentiment_word)
FROM product_reviews;

... 
INSERT INTO TABLE ${hiveconf:RESULT_TABLE}
SELECT extract_sentiment(pr_item_sk,pr_review_content) AS (pr_item_sk, review_sentence, sentiment, sentiment_word)
FROM (SELECT pr_item_sk,pr_review_content FROM product_reviews DISTRIBUTED BY length(pr_review_content)
) pr
```

- Data is generated randomly
- Some map tasks have to process far more data
- Caused typical data skew issue that the query time is determined by the slowest task
Tuning Spark SQL - Executor Number

- **Detecting**
  - Big-Bench on Spark SQL Load phase show slow performance and lower CPU utilization

- **Solution**
  - Increase executor number to improve the resource utilization (e.g. CPU)

![Load Time Chart](chart1)

- **Load Time Chart**
  - Load Time (m) (Lower is Better)
  - 12 executors: 68
  - 192 executors: 19.5
  - Speedup: x3.49

![CPU Usage Chart](chart2)

- **CPU Usage Chart**
  - Lower CPU utilization
  - 12 executors
  - 192 executors
  - Higher CPU utilization
Tuning Spark SQL - JOIN

• Detecting
  • Most workloads in Big-Bench has JOIN & Left Semi Join operation - Join a Large table (Fact Table) with a Small table (Dimension Table)
    ➢ Join(Inner Join): Q5, Q16, Q22, Q26, Q29
    ➢ Left Semi Join: Q9, Q11, Q12, Q15, Q18, Q19
  • ShuffledHashJoin - Shuffle(Disk IO, Network etc.), Uneven sharding, Limited Parallelism

• Solution
  • BroadcastHashJoin (AKA Map Join) – broadcast the small RDD to all worker nodes.
  • Tune “spark.sql.autoBroadcastJoinThreshold ” to enable the broadcast Join

```
SELECT *
FROM inventory inv
JOIN (
  SELECT
    i_item_id, i_item_sk
  FROM item *
  WHERE i_current_price > $(hiveconf:q22_i_current_price_min)
    AND i_current_price < $(hiveconf:q22_i_current_price_max)
) items
ON inv.inv_item_sk = items.i_item_sk
JOIN warehouse w ON inv.inv_warehouse_sk = w.warehouse_sk
JOIN date_dim d ON inv.inv_date_sk = d.d_date_sk
WHERE datediff(d.date, '$(hiveconf:q22_date)' ) >= -30
AND datediff(d.date, '$(hiveconf:q22_date)' ) <= 30
) Q22_coalition_22
```

```
SELECT *
FROM inventory inv
JOIN (SELECT
    i_item_id, i_item_sk
  FROM item *
  WHERE i_current_price > $(hiveconf:q22_i_current_price_min)
    AND i_current_price < $(hiveconf:q22_i_current_price_max)
) items
ON inv.inv_item_sk = items.i_item_sk
JOIN warehouse w ON inv.inv_warehouse_sk = w.warehouse_sk
JOIN date_dim d ON inv.inv_date_sk = d.d_date_sk
WHERE datediff(d.date, '$(hiveconf:q22_date)' ) >= -30
AND datediff(d.date, '$(hiveconf:q22_date)' ) <= 30
) Q22_coalition_22
```

Software and Services
System Technologies and Optimization
Detecting
- Tasks take long time to complete.
- Some tasks OOM
- Lost spark executors

Solution
- Tune “spark.sql.shuffle.partition”
- Too small partition number may cause OOM
- Too large partition number may cause performance degradation.
Other Tunings

- Spark Serializer
- File Format
- Compression
- OS Tuning
- JVM Tuning
- Etc...

Please contact us for more information.
Back up
Enabling Big-Bench on Spark SQL

Enabling Spark SQL 1.2.0 & Spark SQL 1.4.0

- Spark SQL 1.2.0 applying for SPARK-5202,SPARK-5237,SPARK-5364 ,SPARK-4693
- Spark SQL 1.4.0 applying local patch for SPARK-7119

Configuring Big-Bench Spark SQL engine

Topic: How to use BigBench on Spark SQL in Google Group - Big Data Benchmark for Big-Bench
No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

© 2015 Intel Corporation.