Viewing email #32207a87cd3560513e211da9c6e50365897f75db... (and replies):

Click to view as flat thread, sort by date

View Source Permalink Reply

From: Hyunsik Choi <h...@apache.org>
To: tajo-dev <d...@tajo.incubator.apache.org>
Subject: [Discussion] Yarn-related parts and the DAG framework Refactoring
Date: 2013/05/02 19:30:47
List: dev@tajo.apache.org

Hi folks,

I'm going to share the current status of Tajo's DAG framework. Then, I'd
like to discuss the refactor of Yarn-related code and Tajo's DAG framework.
I'm looking forward to some advice and ideas. After this discussion, I hope
that we create some concrete Jira issues.

= Current status for DAG framework =

* A DAG framework consists of two parts: representation and control parts.
* MasterPlan and ExecutionBlock belong to the representation part.
* Query and SubQuery belong to the control part
* Query is a finite state machine and controls A DAG of ExecutionBlocks.
* SubQuery is a finite state machine and controls an ExecutionBlock.

= The below is more detailed description. =

* a distributed execution plan (MasterPlan.java) is a directed acyclic
graph, where each vertex is an ExecutionBlock and edges represents data
channels.

* an ExecutionBlock is a logical execution unit that could be distributed
across nodes.
  ** It's similar to map or reduce phase in MapReduce framework.
  ** an ExecutionBlock includes a logical plan to be transformed to a
physical execution plan that runs on each machine.
  ** a data channel indicates a pull-based data transmission in default and
includes one among repartition types, such as range, hash and list.

* Query internally has a FIFO scheduler (ExecutionBlockCursor) for a DAG of
ExecutionBlocks.
  ** For each call of ExecutionBlockCursor::nextBlock(), it retrieves an
ExecutionBlock to be executed in a postfix order. So, it keeps the
dependency of ExecutionBlocks.

* For each execution block, a SubQuery launches containers and then reuses
them for all tasks of this SubQuery. After all tasks are completed,
SubQuery kills all containers by invoking ContainerManager.stopContainer().

= Discussions =

* FIFO scheduler is inefficient. Even though there are available resources
in a cluster, it executes one ExecutionBlock at a time. We need a better
scheduler.

* For each ExecutionBlock, a SubQuery asks containers to RM of Yarn.
However, I haven't found out nice ways for determine the number of
containers and proper resources for each containers.

* We need a Local Mode (TAJO-45) for Tajo cluster. However, it looks
somewhat complicated because many parts are tied to Yarn. How about
refactoring all parts to be independent from Yarn?

* In the current implementation, Tajo uses Yarn as a cluster resource
manager and launches containers when a query is issued. However, this
approach is very slow. The initialization cost (for allocating and
launching containers) takes at least 3-5 seconds even in 32 cluster nodes
according to my experiences. How about considering standby mode?
  ** Standby mode means that a number of TaskRunners are in standby
according to user's request.

Best regards,
Hyunsik Choi

View Source Permalink Reply

From: Jihoon Son <g...@gmail.com>
Subject: Re: [Discussion] Yarn-related parts and the DAG framework Refactoring
Date: 2013/05/06 04:06:56
List: dev@tajo.apache.org

Hi, Hyunsik

Thanks for sharing the current status of DAG framework.
Current architecture is designed well for the distributed query processing.
Separation of DAG framework into representation and control parts simplify
the implementation and improve the readability.

Your discussions include various kind of issues. I append my opinion as
follows.

1) Scheduling is very important. A better scheduler better than FIFO must
be implemented. As we talked before, maintenance of available
ExecutionBlocks is a good option.

2) Choosing the proper number of containers and resources for each
container is very hard problem. We need to study on this issue.

3) I totally agree with re-factoring all parts to be independent from Yarn.

4) The standby mode will be a good approach to reduce the response time.
However, I have doubts that users can decide the number of TaskRunners run
in the standby mode.

Jihoon

2013/5/3 Hyunsik Choi <hy...@apache.org>

--
Jihoon Son

Database & Information Systems Group,
Prof. Yon Dohn Chung Lab.
Dept. of Computer Science & Engineering,
Korea University
1, 5-ga, Anam-dong, Seongbuk-gu,
Seoul, 136-713, Republic of Korea

Tel : +82-2-3290-3580
E-mail : jihoonson@korea.ac.kr