Stream Processing with Bigdata by SSS-MapReduce

Hidemoto Nakada, Hirotaka Ogawa, Tomohiro Kudoh

AIST (National Institute of Advanced Industrial Science and Technology)

Abstract

We propose a MapReduce based stream processing system, called SSS, which is capable of processing stream along with large scale static data. Unlike the existing stream processing systems that can work only on the relatively small on-memory data-set, SSS can process incoming streamed data consulting the stored data. SSS processes streamed data with continuous Mappers and Reducers, that are periodically invoked by the system. It also supports merge operation on two set of data, which enables stream data processing with large static data.

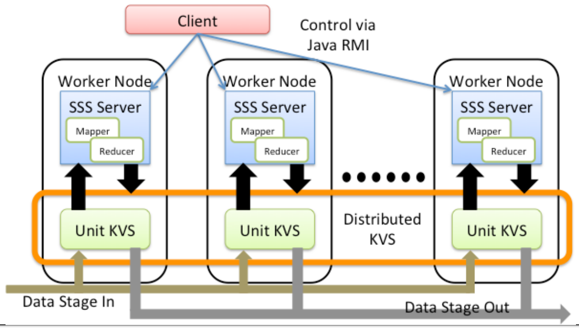
SSS-MapReduce

Distributed KVS based

SSS-MapReduce utilize Distributed KVS (Key-value store) instead of file system. This design simplifies the structure and enables easy adoption of the different computation model, such as Stream Processing.

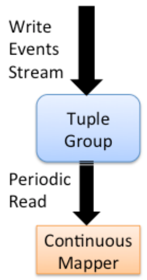
Tuple Group and Owner Compute Rule

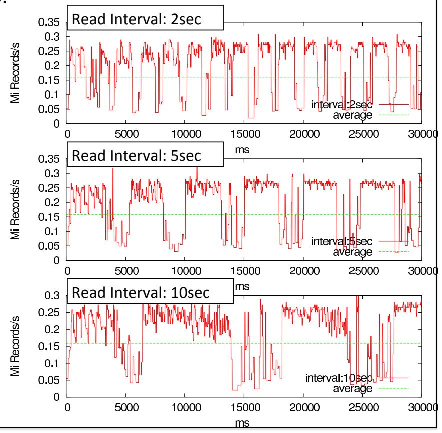
Tuple Groups contain Key-Value pairs and serve as the basic data unit for SSS. Mappers and Reducers read K-V pairs from Tuple Groups and write to other Tuple Groups. Mappers and Reducers run on SSS Server. A SSS Server is responsible for the data on the same node.



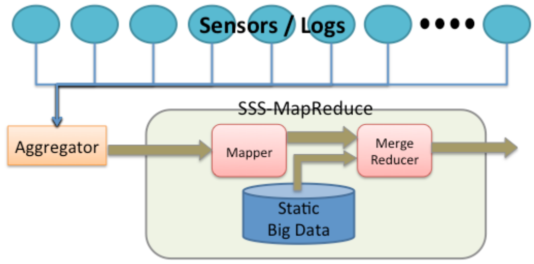
Preliminary Evaluation

We setup invocation interval of continuous Mapper 2, 5, 10 sec., while the event producer keep writing key-value pairs. The Mapper's read operation interferes and degrades the write throughput as show in the graph. The average throughput is around 0.16M events/s.





Stream Processing with SSS



Continuous Mapper/Reducer

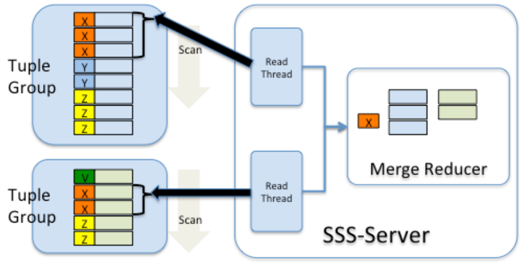
We implemented streamed data processing by invoking Mappers and Reducers continuously and periodically. The interval of periodical invocation can be specified by the user. The Mappers and Reducers reads and delete Key Value Pairs from the specified tuple Group, to ensure that one Key Value Pair is not processed more than once.

To avoid read operation interference in write operation, the Data Server rotates database file for Tuple Groups. When Periodic Mapper or Reducer kicks in on a Tuple Group, the Data Server create a new database file and redirect successive write operations to the new file, while serving the old file for read operations from the Mapper or Reducer.

Merge Reducer

The Merge Reducers are special Reducer that can handle inputs more than one tuple groups. Merge Reducers work just like merge sort. The inputs for the Merge Reducers will be a Key and more than two Value lists. By assigning one input to the stream input buffer, we can describe algorithms that refers streamed data and static large.

```
void merge(Context         context,
           T0              key,
           Iterable<T1>    values0,
           Iterable<T2>    values1,
           Output<T3, T4>  output);
```



Conclusion and Future work

We have shown that SSS-MapReduce could be easily extended to handle Stream data, demonstrating its flexibility due to the adoption of Key-Value Store.

Our future work includes

- Evaluation with real streaming applications
- Performance inprovement

Acknowledgement

This work was partly funded by the New Energy and Indus- trial Technology Development Organization (NEDO) Green-IT project.

http://sss.apgrid.org

National Institute of Advanced Industrial Science and Technology