A Study of Spatial Big Data Technology

Presented by: Kuldeep Sharma (14305R007)
Guide: Prof. N. L. Sarda
Contents

• What is Big Data?
• Hadoop
• Hadoop-GIS & Spatial Hadoop
• Indexing & Querying Spatial Big Data
• Hadoop Scheduling Algorithms
• Spatial Data Mining
• Way Ahead
What is Big Data?

- A Buzz Word!!
- Generally Refered to big volume of data.
- Doug Laney defined big data with 3 Vs: Volume, Velocity, Variety
- What are the usages of big data?
- Business Intelligence, Public Service Delivery, Disaster/Weather prediction, Medical sciences, social media etc.
Are Existing Systems Sufficient?

- We started with files, then came RDBMS.
- Do you think RDBMSs are efficient enough for processing 'Big Data'?
- We need some new systems right?
- Yes! Various systems already exist :) 
- Map-Reduce, A ground Breaking Technology.
- Apache Hadoop is open source project which implements map-reduce.
Spatial Data and why it is so special?

- Multi-Dimensional, objects are points, lines, polygons, other shapes or satellite images, medical images.
- Spatial queries are like kNN, containment, intersect.
- These are very compute intensive.
- Many implicit properties of data like continuity of attribute in space etc.
- No special support for big data in general big data processing systems. Thus, reduced performance.
- No problem! Developed and developing :)
Hadoop

HADOOP 1.0

Applications

MAP-REDUCE

HDFS

HADOOP 2.0

TEZ & Applications

YARN

HDFS
Map-Reduce
YARN (Hadoop 2.0)
Hadoop-GIS

- Scalable and efficient in running spatial queries.
- Uses spatial partitioning and indexing techniques.
- Handles boundary object problem.
- Consists of customizable query processing engine (RESQUE).
- Integrated with hive.
Hadoop-GIS Architecture

Data Acquisition
- Images
- Tables
- Other

Storage
- Data Partitioning
- SHAPES
- FEATURES
- HDFS

Query System
- QL Query Language
  - Cmd Interface
  - Web Interface
- Query Translation
  - Hive query Parser/Query Translator
- RESQUE
  - Spatial Index builder
  - Spatial Query Processor
  - Boundary Handling

Hadoop
Hadoop-GIS Query Processing

- Data: Raw Spatial Data and Spatial Query
- Result: Result of spatial query
- initialization;
- A. Data Space partitioning ;
- B. Uid Allocation and storage in HDFS ;
- C. Pre-processing Query ;
- D. while tile remaining unprocessed do
  - Create Local indexes for each tile;
  - Map Reduce function on each tile in parallel ;
- end
- E. Boundary object handling ;
- F. Post-processing query;
- G. Aggrigation;
- H. Write back result into HDFS ;
SpatialHadoop

- Other spatial big data processing systems use hadoop is black box.
- This property generated various limitations:
  - Hadoop treats spatial data similar to non-spatial data so treating hadoop as black-box will generate same limitations.
  - Map and Reduce functions cannot access the spatial index constructed. So only some specific spatial functions can be defined.
  - Only a limited indexes like Grid indexes are supported by existing systems like Hadoop-GIS etc.
  - Spatialhadoop pushes spatial data support inside hadoop.
SpatialHadoop

Spatial Operations

Query Results

Spatial Parameters

Spatial Queries

Language

Config. Files

Compiled Mapreduce Program

Configured MR Job

MapReduce

SpatialfileSplitter

Spatialrecord Reader

index Info.

Storage

Grid file, R tree, R+ tree etc.

Slaves

Master

Storage/Processing Nodes

Mapreduce Task

File data
SpatialHadoop

- In Language layer, It adds an language (pigeon) with spatial data types and operations.
- In Storage layer, It makes two layer index using widely used spatial indexing techniques like R tree or Grid file.
- In Mapreduce layer, It adds two more components, SpatialFileSplitter and Spatial-RecordReader for efficient spatial data processing.
- In Operation layer, Spatialhadoop have various operation including range, kNN etc. Other operations can also be implemented by users.
Indexing & Querying

- Spatial indexing techniques try to exploit implicit relationships.
- Very important because query performance depends on speed of retrieval of data.
- Various indexing techniques like R-tree, grid file etc.
- These are designed for procedural programs, but map-reduce is functional programming.
- Two level indexing: Global Index and Local Index.
- Various indexing algo. Can be used for global and local indexing.
Indexing & Querying

- Tree based methods are popular because they provide fast traversal but for these, we need some type of ordering of data.
- Space filling curves like hilbert curve or z-index are used for this purpose.
- Traditional systems: space driven or data driven. New techniques use combination of both.
- Non-leaf node contain very less info to keep global index size small so that it can fit in main memory for fast access.
Indexing and Querying

- Hybrid Index: Both spatial and non-spatial data. Combination or R-tree and inverted file index.
- R-tree for space traversal then from leaf node of r tree, traverse inverted file index.
- Multi-dimensional Index: two popular queries; kNN and range. Can use multiway r tree, grid files.
- Linearization
Indexing and Querying

• Two step query processing: Filter and Refine.
• Filter Step: In this step, the spatial predicate of query is checked with MBR of the objects. To speedup the process various kinds of indexes are used. This step provides us approximate result set.
• Refinement Step: In this step, each object returned by Filter Step is checked against spatial predicate of the query and actual result is produced.
Plan Generation

• Initially a logical plan for single machine execution is generated. Then in second step, this plan is extended for distributed query execution.

• **Spatial Selection:**
  • A sequential search if no index else index is used to create a much efficient R-tree based search plan.

• **Spatial Join:**
  • Depends on the indexes available with us.
Plan Generation

- **Overlap Join:** This is the plan when both relations are indexed on join attribute. In this case, first overlapping partitions of both relations are determined using MBRs of both relations objects and then, the objects with overlapping partitions are shuffled to one node and then local indexing is performed.

- **Partition Join:** This is the plan when only one relation say R is indexed. In this case, the S is partitioned to match the index of other table. Then similar way, overlapping partitions are shuffled and local spatial join is perform on these.

- **Co-Partition Join:** This is the plan when no table is indexed. In this both files are partitioned using common grid and then on corresponding matching grids of both files, spatial join is performed in distributed manner.
Query Execution on MapReduce

- **Spatial Selection:** Only map task required.

```
Input File (HDFS)
<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Split</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Record Reader</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Map</td>
</tr>
<tr>
<td>(Point or Range)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>HDFS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Input Split</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Record Reader</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Map</td>
</tr>
<tr>
<td>(Point or Range)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>HDFS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Input Split</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Record Reader</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Map</td>
</tr>
<tr>
<td>(Point or Range)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>HDFS</td>
</tr>
</tbody>
</table>
```
Query Execution on MapReduce

- **Spatial Join**: Both Map and Reduce tasks required.

![Diagram showing the process of Spatial Join on MapReduce](image)
Query Execution on MapReduce

- **KNN**: Just one map-reduce job required. In map, distances between query point and every object are calculated. The reduce function then sorts the distances and outputs first k objects.
- Require just 1 reduce task.
- A more efficient algo. Uses two map-reduce jobs. First map-reduce job just finds NN in same partition only.
- Second map-reduce job finds NN for those remaining objects whose NNs were not returned in first job.
- **Hybrid Query Processing**: System that can process spatial as well as non-spatial data. Find the query type and then create efficient plan accordingly.
Hadoop Scheduling Algo.

- **User Level Algorithms:** In these types of algorithms, a user queue is maintained and one user is selected to schedule its jobs.

- **Job Level Algorithms:** These algorithms selects a job from queue of jobs. There can be a single queue for all users or separate queues for each user.

- **Task Level Algorithms:** These algorithms schedules a task of selected user and job. Map-Reduce has two types of tasks map and reduce. Hadoop also runs speculative tasks for slow tasks.
User Level Algo.

• **Fair Scheduler**: It tries to give fair share to every user.

• It does not block small jobs of interactive users because of large batch jobs of some users.

• **Capacity Scheduler**: It divides resources among the user queues.

• It puts a limit on number of tasks performed by one user and thus, provides minimum capacity to each user.
Job Level Algo.

- **FCFS, FAIR**
- **Hybrid Scheduler:** Combination of FCFS, SJF, smallest job first.
  - Next job = \((\text{wt/avg. Wait. time})^a \times (\text{mt/ Avg. Map. time})^b \times (\text{nR/ Avg. No. R task})^c\)
- **Delay Scheduling:** Head of line and sticky slot problems with fair scheduler.
  - Strict queue pattern does not give us much opportunity to schedule a task on node with local data.
  - Try to schedule a task on node with local data without incurring much cost or delay.
  - But this head of line job can’t be skipped for much time. So we keep a skipcount. If skipcount exceeds threshold then schedule non-local tasks.
Task Level Algo.

- Hadoop handles poorly performing nodes called 'stragglers' by running speculative copy of tasks running on this node, on another node.
- Task selection for speculative execution is based on comparison of task progress rate with avg. Task progress rate of all running tasks.
- This heuristics not fit for heterogeneous environments like data centers etc.
- **Longest Approximation Time to End (LATE):** run speculative copy of task for which we assume that it will finish farthest in future.
- Speculative tasks can’t be scheduled on slow nodes. Also can’t run too many speculative tasks at a time.
Task Level Algo.

- **Self Adaptive MapReduce Scheduling Algorithm (SAMR):** Problem with LATE is. It calculates progress of tasks are computed in static manner.

- In SAMR, task progresses are calculated dynamically and changes in environment are adapted by it.

- It reads historical information on each node and accordingly change weighs of various stages of map and reduce tasks.

- Historic information is updated after every execution if a task on node.

- SAMR calculates two progress rates of nodes, one for map tasks and another one for reduce tasks
Task Level Algo.

- **Center of Gravity Reduce Task Scheduling (CoGRS):** Hadoop doesn’t exploit data locality and address partitioning skew while scheduling reduce tasks.
  - High amount of traffic.
  - CoGRS take into account data locality and partition skew and schedules reduce tasks at center-of-gravity node.
  - The Center-of-gravity node is calculated using weighted total network distance.
  - The center-of-gravity node is the one, which provides least WTND.
Spatial Data Mining

• Mining is all about relations or Pattern finding.
• Patterns can be used for route planning, congestion control, hotspot detections, defence technologies etc.
• Due to complexity of data, spatial relationships etc, mining spatial data is more difficult as compared to numerical data.
• **Spatial Co-location:** those patterns which appear in closed geographic proximity.
• Two types of algorithms: spatial statistics based and association rule mining based.
• Map-reduce based algo. first partitions space. Map tasks parallely search for co-locations patterns in their respective partitions. In reduce step, various patterns are combined to form final results.
Spatial Data Mining

- **Spatial regression & classification:** These methods are used to find the relations between independent and dependent variables.

- While performing spatial regression or classification, we should also take into account the relationships among spatially co-located points.

- Markov Random Field (MRF) based classifier, Logistic Spatial Autoregression (SAR) model, Geographically Weighted Regression (GWR).

- Various map-reduce based algo. Are also available like MapReduce bases Neural Network etc.
Spatial Data Mining

- **Spatial Clustering**: process to group spatial objects which are more similar to each other and dissimilar to objects in other groups.

- Hierarchical Clustering, Partitional Clustering, Density Based Clustering.

- Various algorithms try to parallelize the clustering algorithms using MapReduce like parallel K-Means etc.

- **Hotspot Detection**: HotSpot is also a kind of highly active cluster. Spatial correlation among the objects within the hotspot region is very high but decreases exponentially at the boundary.

- Clustering algo. Extended to find hotspots or application specific algo. Like STAC (Spatio Temporal Analysis of Crime).
Way Ahead

• Parallel geometric Algorithms using map-reduce which can be more efficient.
• Locality aware delay scheduling of reduce task for less network congestion.
• Spatial property aware scheduling policy.
• Various spatial data mining algorithms implementation using map-reduce.
• Data location in Rack or data-center aware indexing.
• Deep Learning algorithm for spatial data implementations using map-reduce.
• Big time-series database management systems.
• High Performance Architectures for big data query executions.
• Thank You :)