Designing K-Means for the cloud

Matthieu Durut, Fabrice Rossi

March 6, 2012
Few reminders about distributed K-Means

\[ K\text{Means \ Sequential \ Time} = (3Knd + Kn + Kd + nd) \ast I \ast T^{\text{flop}} \]
\[ \simeq 3Knd \ast I \ast T^{\text{flop}} \]
KMeans SMP Distributed Cost = \( T^\text{comp}_P \)
\[
= \frac{(3Knd + Kn + Kd + nd) \times I \times T^\text{flop}}{P}
\]
\[
\approx \frac{3Knd \times I \times T^\text{flop}}{P}
\]
KMeans DMM Cost

\[ DMM\ Cost \]

\[ T_P^{comp} + T_P^{comm} \]

\[ = \frac{(3Knd + Kn + Kd + nd) \times I \times T^{flop}}{P} \]

\[ \approx \frac{3Knd \times I \times T^{flop}}{P} + O(\log(P)) \]
Figure: Merging centroids logic
classical issue: pipes

The size of the biggest pipe (on top of the tree) and the aggregated bandwidth.
modelisation of $T_P^{comm}$ in DMM

\[
T_P^{comm} = \left\lceil \log_2(P) \right\rceil \sum_{p=1} \left( 2 \frac{IKd \times SoD}{B_{DMM, MPI}^P} + 5IKd \times T^{flop} \right)
\]
modelisation of $T_P^{comm}$ in DMM (2)

$$T_P^{comm} = \left(2 \frac{IKd \ast SoD}{BP_{avg} \ast DMM \ast MPI} + 5IKd \ast T^{flop}) \ast \left\lceil \log_2(P) \right\rceil$$
\[
\text{Speedup} = \frac{\text{KMeans Sequential Cost}}{\text{KMeansDMMDistributedCost}}
\]

\[
= \frac{\text{KMeansSequentialCost}}{T_P^{\text{comp}} + T_P^{\text{comm}}}
\]

\[
= \frac{3nKdIT^{\text{flop}}}{3nKdIT^{\text{flop}}_P + \left(2 \frac{IKd*SoD_{B_{DMM, MPI}}}{B_P} + 5IKdT^{\text{flop}}\right) \lceil \log_2(P) \rceil}
\]

\[
= \frac{3nT^{\text{flop}}}{3nT^{\text{flop}}_P + \left(2 \frac{SoD_{B_{DMM, MPI}}}{B_P} + 5T^{\text{flop}}\right) \lceil \log_2(P) \rceil}
\]
Some comments

- RAM is not modelize ⇒ speed up might be greater
- speedup independant of K, l and d
- winner paper model!

\[
\lim_\limits_{n \to +\infty} \text{SpeedUp} = P
\]
First case study: EDF

- \( n = 20\,000\,000 \) time series
- \( d = 24 \text{ hours} \times 365 \text{ days} \times 10 \text{ years} = 87600 \) values per series
- \( K = \sqrt{n} = 4472 \) clusters
- Note: volume of timeseries is 12.3 TeraBytes.
37 years
Yahoo terasort winner

- 910 nodes
- 4 dual core Xeons @ 2.0ghz per a node
- 1 gigabit ethernet on each node
- 40 nodes per a rack
- 8 gigabit ethernet uplinks from each rack to the core
\[ T_P^{\text{comp}} = \frac{(3nKd + nK + dK + nd) \times I \times T^{\text{flop}}}{P} \]
\[ = 322893 \text{seconds} \]

\[ T_P^{\text{comm}} = (2 \times \text{Broadcast time between 2 processors} + \text{Merging time}) \times \lceil \log_2(P) \rceil \]
\[ = (2 \frac{IKd \times SoD}{B_{DMM,MPI}^{P,avg}} + 5IKd \times T^{\text{flop}}) \times \lceil \log_2(P) \rceil \]
\[ = 225866 \text{seconds} \]
Yahoo terasort winner(2) bounded by aggregated bandwidth

\[ B_{P,\text{avg}}^{DMM,\text{MPI}} = 109 \text{Mbits} \]

\[ T_P^{\text{comm}} = (2 \ast \text{Broadcast time between 2 processors} + \text{Merging time}) \ast \lceil \log_2(P) \rceil \]

\[ = \left( 2 \frac{IKd \ast SoD}{B_{P,\text{avg}}^{DMM,\text{MPI}}} + 5IKd \ast T^{\text{flop}} \right) \ast \lceil \log_2(P) \rceil \]

\[ = 434155 \text{seconds} \]
Pricing on Azure

- Storage price: 1845 dollars/Month (0.15 dollars/GB/Month)
- No charge for internal transfers?
- Update: 10 GBytes/Month: 3 dollars (0.3 dollars/GB)
- Cost of running the KMeans: (very rough) 0.12 dollars/hour * 324120 = 40 000 dollars
- Storage requests: ? from 0.01 dollars to 11 millions dollars
Figure: Paas, Iaas, SaaS
Paas and Iaas: some tradeoffs

- abstraction/control tradeoff
- scalability/cost to develop tradeoff
Introducing Lokad.Cloud

- higher level abstractions
Azure Abstractions

- Azure Queues
- Azure BlobStorage
- Azure TableStorage
- Azure SQLStorage
Azure Queues

- loosely coupled asynchronous communication mechanism
- not scalable yet (in size and number of messages)
- FIFO when not sought
- messages can be processed several times if worker fails (see idempotency)
- objects up to 50 GB
- consistency?
- Key/value system
Azure Table Storage

- Tables/Entities/Properties
- key/value pair also
Azure Queue Services

- SOA
- Queue consumers
ACID/BASE constraints on Cloud

- ACID
- BASE
- CAP Theorem
- CLAP system
ACID/BASE constraints on Azure

- Availability: No
- Internet and Electricity: neither
- Consistency: ?
Consistency on Azure: replicas and consistency

- Block commits
- Etag
- very few overwrites
Consistency on Azure: transactionalit

• The SQL trick
Azure tradeoffs choices

- read/write tradeoff
- Responsivity/efficiency
- Synchronous replication/Asynchronous replication?
BlobStorage or TableStorage?

- Partition Key and locality
- Size of objects
- Efficiency of Blobstorage for objects larger than 4Kbytes
- Price
Abstractions performances

- BlobStorage Read: 13MBytes/sec/worker. Aggregated bounded to 393 MBytes/sec
- BlobStorage Write: 5MBytes/sec/worker. Aggregated bounded to 124 MBytes/sec
- Direct worker communication: 55MBytes/sec on very large messages
- VM: from 1 to 8 cores
- Queues throughput: 500 messages/sec
Redesign of the algorithm

- SOA
- No Master
- All workers can be switched
- No worker/data per affinity but ...
- Scaling up as an initiative in PaaS
- No exact workers count
- Pinging queues as a tradeoff between cost and simplicity
- Queue Messages count
- idempotent jobs
Redesign of the algorithm (2)

- Idempotent Counters
- Workers communication?
Proposed Algorithm

Scheme
Difficulty to anticipate speedUp

- variability of the bandwidth
- latency due to pinging queues
- a worker can fail, a VM can be killed
- Storage unavailability
- Stragglers issues ?
Is time to load data important?

\[
T_{Load}^P = \frac{nd \times SoD}{P \times \max(Bandwidth_{read}, \frac{MaxReadThroughput}{P})}
\]

\[
T_{Load}^P \ll T_{Comp}^P
\]

Implicit supposition: loading is run once.

\[
\frac{SoD}{3IK \times T_{flop} \times \max(Bandwidth_{read}, \frac{MaxReadThroughput}{P})} \ll 1
\]
Communication modelisation

\[ T_{P}^{\text{comm,periteration}} = \text{Time to read 1 blob "simultaneously" by } P \text{ workers} + \text{Time to write } P \text{ blobs "simultaneously"} \]
\[ + \text{Time to read } P \text{ blobs by 1 worker} + \text{Time to merge the different centroids version} \]
\[ + \text{Time to write 1 blob per 1 worker} \]
\[
T_{\text{comm, per iteration}}^P = \frac{Kd \times SoD}{\text{Min}(\text{Bandwidth}^{\text{read}}, \frac{\text{BlobReadThroughput}}{P})} + \frac{Kd \times SoD}{\text{Min}(\text{Bandwidth}^{\text{write}}, \frac{\text{MaxWriteThroughput}}{P})} + \frac{PKd \times SoD}{\text{Bandwidth}^{\text{read}}} + (P - 1)Kd \times 5\text{floatOperations} \times T^{\text{flop}} + \frac{Kd \times SoD}{\text{Bandwidth}^{\text{write}}}
\]
$T_P^{\text{comm}} \sim \frac{IPKd \ast SoD}{\text{Bandwidth}^{\text{read}}}$
SpeedUp

\[ \text{SpeedUp} = \frac{T_{1}^{\text{comp}}}{T_{P}^{\text{Comp}}} + T_{P}^{\text{comm}} \]

\[ \geq \frac{3IKdnT^{\text{flop}}}{3IKdnT^{\text{flop}}/P} + \frac{IPKd \ast \text{SoD}}{\text{Bandwidth}^{\text{read}}} \]

\[ \geq \frac{3nT^{\text{flop}}}{3nT^{\text{flop}}/P} + \frac{P \ast \text{SoD}}{\text{Bandwidth}^{\text{read}}} \]
Optimal Number of workers

\[ P^* = \sqrt{\frac{3nT^{\text{flop}} \cdot \text{Bandwidth}^{\text{read}}}{\text{SoD}}} \]
Numerical Example

- $n = 432\ 000$
- $K = 657$
- $d = 168$
- $I = 100$
- $P_{\text{Max}} = 64$
- $\text{Bandwidth}^{\text{read}} = 20\ \text{MB/sec} (\text{resp.} 5\ \text{MB/sec})$
Results

Sequential KMeans time = 14300 seconds

Optimal number of workers = 57 (resp 28)

SpeedUp = 28.5 (resp 14)
Improvements: more reducers?

scheme
Direct worker communication?