Next generation data storage system to support big data, IoT and machinelearning at the Norwegian Meteorological Institute

Isak Buhl-Mortensen[1], Søren R. Jørgensen[1], Thomas N. Nipen[1], Ivar A. Seierstad[1], and Cristian Lussana[1]

[1]the Norwegian Meteorological Institute

November 22, 2022

Abstract

The Norwegian meteorological institute (MET Norway) routinely collects and archives in-situ observations measured by conventional weather stations following the WMO standard. However, it is apparent that non-conventional observations, those shared by private companies and citizens, cannot be ignored. Over the last couple of years, the number of such observations has constantly risen. From the point of view of a national meteorological service,this data comes with a number of issues, such as insufficient metadata and a total lack of control on both the measurement practices applied and the instrumentation used. On the other hand, this large volume of non-conventional data (up to hundreds of observations per km square per minute) allows for the near-surface atmospheric state to be observed at an unprecedented level of detail thus opening new possibilities for disaster risk reduction and research in atmospheric sciences.Redundancy is the key factor that helps transform otherwise unreliable data into usable data for national meteorological services. MET Norway has recently improved the temperature forecasts on Yr.no by introducing amateur station data into the processing chain. Yr.no has millions of users per week so this improvement is beneficial fora large community. This has required a tailored system based on two aspects: (1) distributed storage and (2) data quality control.We present our plans for a distributed database for mass storage and analysis of in-situ data. This storage backend will lay the foundation for products based on Big data, IoT and machine learning. To match a constant increase in data load, it becomes necessary to scale out and embrace the nature of distributed systems - a constant compromise between performance (availability) and information consistency. We favor availability at the expense of eventual consistency (milliseconds). For transactions that require higher consistency, distributed database management systems like Cassandra (C*) allow clients to specify the level of consistency. C* also supports ordered columns and a time-window compaction strategy, making C* performant for time series data. In terms of redundancy, a C*cluster employs leaderless replication and therefore has no single point of failure. This makes C* popular - it is the main technology behind Netflix's time series storage solution of customer viewing history.C* may sound perfect for time series data, however there is a need for other access patterns, and unlike a relational database this comes at a cost, denormalization - SQL-like operations such as joins are not supported. To overcome these constraints, we denormalize the data into gridded, time series, and point cloud representations. For the point cloud representation we are currently testing a model that distributes data across the cluster via geohash, and allows for data selection within a geofence.

Preparing for Big Data & IoT @ MET Norway

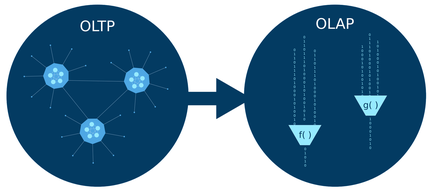
Isak Buhl-Mortensen*, Søren R. Jørgensen, Thomas N. Nipen, Ivar A. Seierstad and Cristian Lussana
Norwegian Meteorological Institute, Oslo, Norway

* Contact : isakbm@met.no

Meteorologisk institutt

OLTP => OLAP

Relational databases have been responsible for both the **transactional** part, as well as the **analytical** part of databases. This does not scale well, and we are seeing a need to differentiate between online transaction processing (**OLTP)** and online analysis processing (**OLAP**). We are currently designing a new OLTP system at MET Norway, designed to support millions of IOPS.



Online Transaction Processing.

- **Low-Latency** writes
- **Highly Available**
- **Latest data** / frequently used
- Stores **GB to TB** of data
- **Row** oriented

... ***Cassandra***

Online Analysis Processing.

- **Bulk** import / event stream
- **Historical data**
- Stores **TB to PB** of data
- **Column** oriented

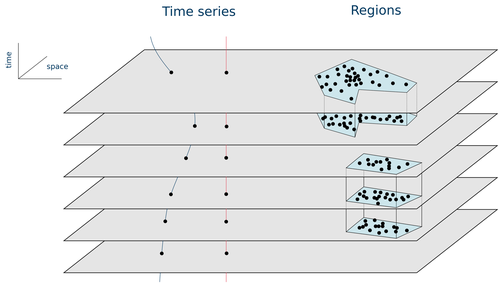
... ***Lustre***

source material : Designing Data-Intensive Applications Martin Kleppmann

Time series & Regions

When designing a data model in a distributed database manager like Cassandra, it is important to model around the queries you expect to be used. This is because having multiple indexes on the same table does not scale well, leading to the mantra "Model around queries".

In our case, queries naturally split into two types, time series and regions. For time series, an epoch counter and a unique identifier can be used as the primary key. This is sufficient in most cases. However it is less clear how to partition (choose a primary key) for regions.



Regional Queries and Geohash

The challenge of picking a good partition key for the purpose of regional queries is solved by using a truncated geohash.

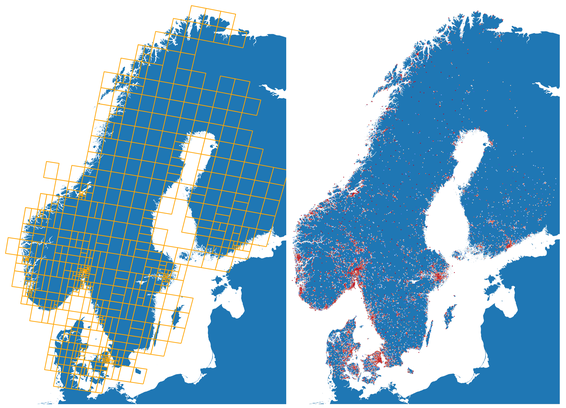
Each observation has a coordinate (ϕ,λ), this is an excellent candidate for a partitioning key, except that it is two dimensional and as such is not suited for filtering in Cassandra. There is a better candidate, a **geohash,** a map from (ϕ,λ) to uint64.

This is the approach taken by Google in their **S2Geometry** library. The image to the right shows how a **Hilbert Curve** wraps the Earth. It's size has been exaggerated, in reality it produces a uint64 that covers the Earth down to a precision of ~ 1 cm. Furthermore there are no issues at the poles. For more information see http://s2geometry.io.

We are in the early stages of developing a partitioning algorithm that makes use of the S2Geometry library. Some promising early results are shown in the figure below. **A**) is a top down view of a single "PartitionTree" generated by ingesting Netatmo Data from Scandinavia, **B**) shows a sample of this data, many times more stations than our own.

The PartitionTree algorithm naturally adapts to regions of higher observation density, as a result Scandinavian capitals are clearly visible in **A**). The partition tree solves two problems: Distributing data evenly across the cluster, and simultaneously reducing query time. It achieves the same kind of complexity as an R-tree would.

Child nodes are created only once needed, so the tree grows it's branches in rhythm with the rate of data ingestion. Furthermore, once the tree surpasses a given size or age, the process can be repeated, but instead of starting from scratch, it may base it's initial structure on what is anticipated.

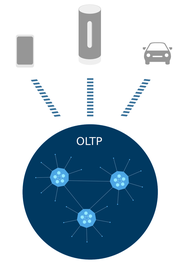


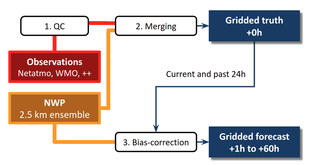
A) Geohash partitioning B) Netatmo stations

IoT Data Improves Forecast

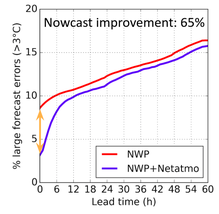
Internet of Things (IoT) data, from mobile phones, cars or other connected devices, is becoming readily available. The question becomes, how to integrate this data into existing pipelines, and what kinds of benefits this can have on weather forecasts.

MET Norway is already using observations from amateur weather stations, such as Netatmo. It is used for model post processing. Even though the quality of the data is lower, the abundance of the data makes up for it. It has reduced the percentage of large errors in nowcasting by 65%, see below.



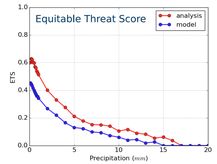


Post processing pipeline

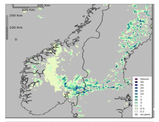


Comparing model and analysis

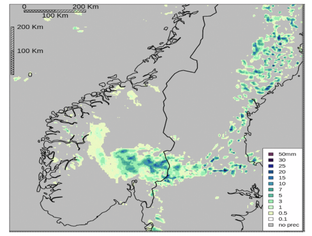
A recent study concerning precipitation, clearly shows improvement there as well. The combination of model and data, the analysis, significantly outperforms the model. The graph to the right scores model and analysis using the Equitable Threat Score. Higher is better, meaning a better likelihood for precipitation larger than Xmm to actually occur.



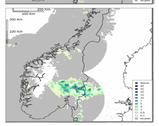
A



B



C



A, B, C show respectively the model, analysis, and data, for precipitation over Norway and Sweden.

Figures on this tab are borrowed from Cristian Lussana's talk "Private observations Improve Met Norway's operational forecasts", based on work that will appear in future publications:

"Lussana C., Seierstad I.A., Nipen T.N., and Cantarello L. "Spatial interpolation of two-meter temperature over Norway based on the combination of numerical weather prediction ensembles and in-situ observations", submitted to QJRMS " and "Nipen T.N, Seierstad I.A., Lussana C., Kristiansen J. and Øystein H. "Adopting citizen observations in operational weather prediction" submitted to BAMS"

Technologies used

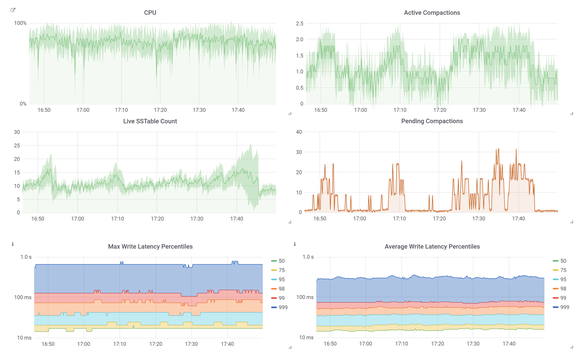
The choice was made to go distributed, as this gives the ability to scale, followed by natural redundancy and flexible hardware choices.

There are several choices for distributed database managers. Cassandra came out among the top in study at the University of Toronto in 2012, entitle "Solving Big Data Challenges for Enterprise Application Performance Management" link : https://arxiv.org/pdf/1208.4167.pdf. In addition we are making use of several other open source projects:

cassandra · GO · Grafana · kubernetes · Prometheus · docker · GRPC · NATS

Cluster Stress Testing

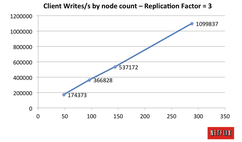
Early testing on a small cluster consisting of 5 nodes. Each node has 8GB RAM, 100GB Spinning Disk, 4 VCPUs @ 2.2 GHz. ~50000 Writes/sec.



In the upper four curves, an envelope is also shown representing the spread across the cluster, (± standard deviation). The load and behavior seems consistent across the cluster, indicating a reasonable choice of partition key.

We expect to outperform this cluster by an order of magnitude once we are ready to run the same tests on the new hardware.

To the right are results that Netflix published for Cassandra performance on AWS. Horizontal axis showing node count. The types of nodes used were M1 Extra Large, which have 4 medium CPU's, 15GB RAM, 4x400GB Disks. The key takeaway is linear scalability.



netflix-techblog/benchmarking-cassandra-scalability-on-aws-over-a-million-writes-per-second-39f45f066c9e