We had a serious task ahead of us - to develop a high-load system capable to collect and store massive amounts of real-time data submitted by oil rig sensors (1 000 - 1 000 000 devices sending data with any interval between 1 and 60 seconds). The end product also had to meet the following major requirements:

- Security is critical.
- Very high throughput.
- Open-source technology stack for critical parts of application.

Architecture

The system architecture is shown on the figure below.Here are the main parts of this complex software:

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk

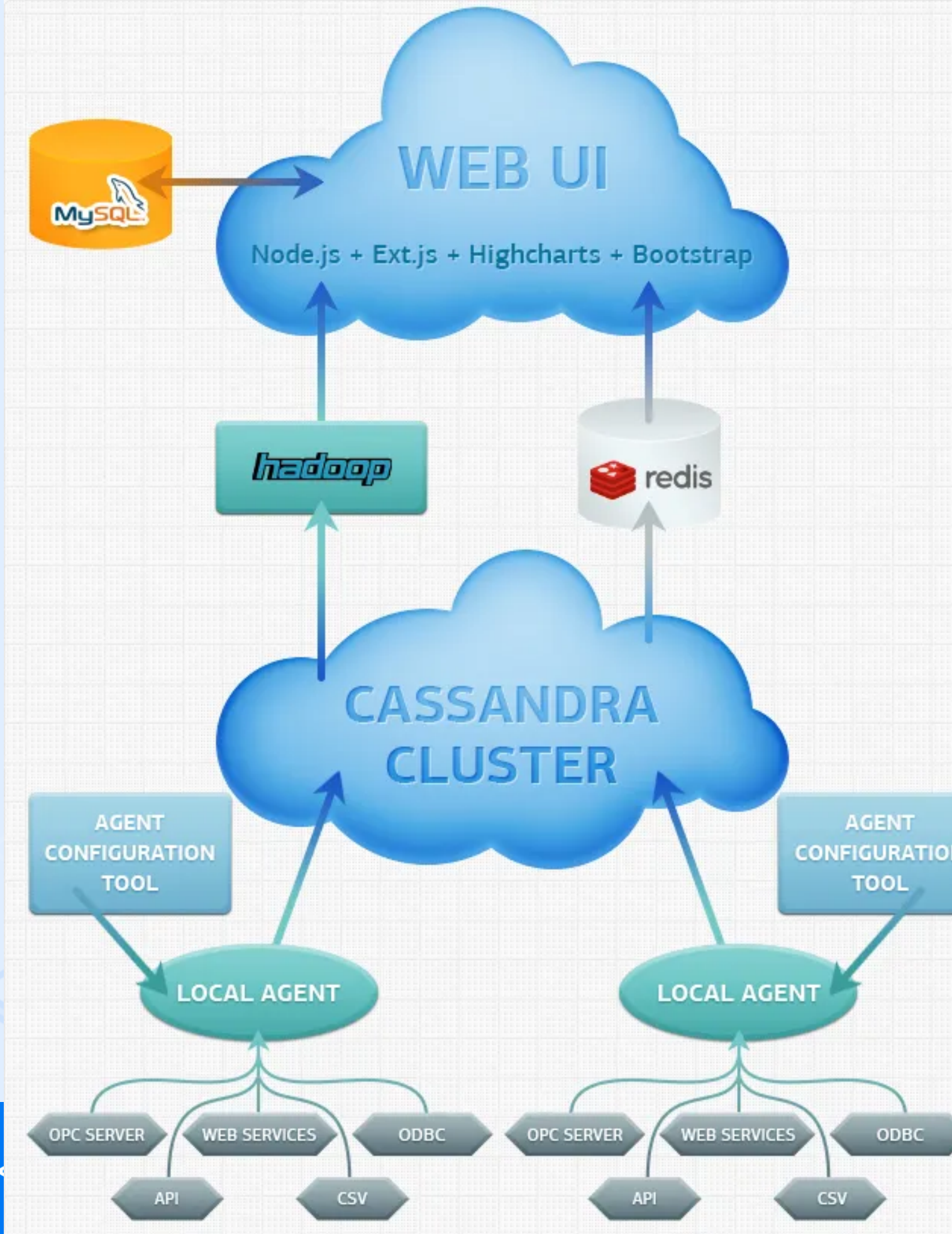
- **Local agent** is a .NET app. It works as a Windows service that connects to various data sources and consumes data from them. Data sources can be OPC-servers, Web Services, ODBC instances, Web API, CSV sources. Each local agent has a unique identifier. Its configuration is defined either by the web part or by a configuration file. Availability of an Internet connection determines the preferable choice here.

- **Cassandra cluster.** Cassandra is an open source NoSQL database management system implemented in Java by the Apache Software Foundation. It's a great solution for handling huge amounts of data. Cassandra provides scalable, decentralised and fault-tolerant data storage. It has its own query language called CQL (Cassandra Query Language) and supports MapReduce technology.

- **Web part** is divided into two parts. The first one is a website with an admin panel and a user interface. Admin can manage local agents and agent settings, manipulate data format, user privileges, etc. Users can log in, view historical data and charts representing real time data. We chose Highcharts for chart rendering, Bootstrap and Ext.js for the UI. It allowed us to develop a fully functioning and good-looking interface really quick. All the backend logic is implemented in Node.js. The second part is a "cache manager" app implemented in Java. It was created for updating and clearing the cache. We opted for Redis as a cache engine.

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk

magora


WEB UI
Node.js + Ext.js + Highcharts + Bootstrap
MySQL
hadoop
redis
CASSANDRA CLUSTER
AGENT CONFIGURATION TOOL
AGENT CONFIGURATIO TOOL
LOCAL AGENT
LOCAL AGENT
OPC SERVER
WEB SERVICES
ODBC
API
CSV
OPC SERVER
WEB SERVICES
ODBC
API
CSV

Implementation details

One of Cassandra's distinctive features is clustering. We built a cluster consisting of four machines. The story of setting up Cassandra deserves a separate article. After configuring and balancing tokens for each node, we ended up with the following diagram in the OpsCenter (a visual management and monitoring tool for Cassandra):

Circles represent the Cassandra cluster nodes. The green colour indicates that all of them are up. Even distribution of the circles denotes that the cluster is balanced.
We implemented the Web UI in Node.js and Express.js framework. Using Express.js allowed us to promptly create a robust Web API in order to separate the backend logic from the UI layer.The user

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk

accounts and settings are stored in MySQL. We utilized a Node ORM2 package to create data access logic. It is a very convenient ORM for Node.js that supports many relational and NoSQL database engines.

To avoid UI lags we created a cache run by the Redis engine. Redis stores the last N records received from local agents. The reasoning behind this decision is that Cassandra may contain millions and millions of records, and it also has known data filtering problems. Now we don’t have to pull data from Cassandra each time, we just get it from Redis.

The web UI contains several line charts and tables with real time data (it is updated every second). We set up an Apache Hadoop for analysing and managing the historical data. Hadoop is a perfect framework for working with big volumes of data. We utilized the Hadoop MapReduce module.

Summary

Why do we need all these technologies? Nowadays, we have to deal with increasing volumes of data that concerns various aspects of life: science, security, social networks, trading, etc. Due to the scalability and schema-free style of Cassandra we can use it to store any type of business models. We can do it in real time and guarantee data security, fault-tolerance and availability at all times. In addition, all of these instruments are free. So you can build high-load apps for your needs in a fast and convenient way.

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk