DAQBroker
General purpose instrument monitoring

António Dias

CENTRA - SIM (FCUL)
CERN - Switzerland

12 July 2018
Motivation
Motivation

- CLOUD experiment (CERN)
  - International collaboration (20+ partner institutes)
Motivation

- CLOUD experiment (CERN)
  - International collaboration (20+ partner institutes)

- Study processes of atmospheric cloud formation
  - Improve climate modeling
  - Discover new cloud formation processes
Motivation

- CLOUD experiment (CERN)
  - International collaboration (20+ partner institutes)

- Study processes of atmospheric cloud formation
  - Improve climate modeling
  - Discover new cloud formation processes

- 30+ instruments a year
  - Constantly rotating set
  - Different formats, sources, etc...
Motivation

- CLOUD experiment (CERN)
  - International collaboration (20+ partner institutes)
- Study processes of atmospheric cloud formation
  - Improve climate modeling
  - Discover new cloud formation processes
- 30+ instruments a year
  - Constantly rotating set
  - Different formats, sources, etc...

This is a big data acquisition challenge!
What is DAQBroker?

- Software framework (package)
What is DAQBroker?

- Software framework (package)
  - Instrument data gatherer
    - 1. Automated operation
    - 2. Collects from various instrument sources
    - 3. Stores ordered data in a central database
What is DAQBroker?

- **Software framework (package)**
  1. Instrument data gatherer
    1. Automated operation
    2. Collects from various instrument sources
    3. Stores ordered data in a central database
  2. Data monitor
    1. Web tools for data access & sharing
    2. Third-party API
What is DAQBroker?

- Software framework (package)
  1. Instrument data gatherer
     1. Automated operation
     2. Collects from various instrument sources
     3. Stores ordered data in a central database
  2. Data monitor
     1. Web tools for data access & sharing
     2. Third-party API
  3. Experiment coordinator
     1. Experiment/Event log
     2. Instrument log
Flexible usage
Flexible usage

- Use on any system
  - OS independent & Open source
  - Python package

<table>
<thead>
<tr>
<th>DAQBroker components</th>
<th>Linux</th>
<th>Windows</th>
<th>Mac</th>
<th>Android</th>
<th>iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Application</td>
<td>✔</td>
<td>✔</td>
<td>✔*</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>API / Instrument/data storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network agent</td>
<td>✔</td>
<td>✔</td>
<td>✔*</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>User interface</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Data visualization/download</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chart creation</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

* Pending testing
Flexible usage

- Use on any system
  - OS independent & Open source
  - Python package
- Database independent
  - Major RDB engines local/remote (SQLAlchemy)
Flexible usage

- Use on any system
  - OS independent & Open source
  - Python package
- Database independent
  - Major RDB engines local/remote (SQLAlchemy)
- Standalone or network automated data collection
  - Client/Server communication architecture
Instrument model

- 3-step Virtual Instrument Representation
Instrument model

- 3-step Virtual Instrument Representation

1. General information
   - Name, information, operator, ...

Instrument

António Dias (amcbd89@gmail.com)
Instrument model

- 3-step Virtual Instrument Representation

1. General information
   - Name, information, operator, ...

2. Data source information
   - Type of data, time provided, ...

Instrument

Data Sources

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:/data/home/user/data</td>
<td>Port: 9000</td>
<td>COM1 / ttyS0</td>
</tr>
</tbody>
</table>
Instrument model

- 3-step Virtual Instrument Representation

1. General information
   - Name, information, operator, ...

2. Data source information
   - Type of data, time provided, ...

3. Data parsing information
   - Individual data stream information (name, order, type...)

Instrument

Data Sources

S1: C:\data
/home/user/data

S2: Port: 9000

S3: COM1 / ttyS0

Data Channels

António Dias (amcbd89@gmail.com)  DAQBroker - CHEP 2018  12 July 2018
Performance

Low Computing Power (LP)
- CPU: 4× ARM Cortex-A53, 1.2GHz
- RAM: 1GB LPDDR2 (900 MHz)
- Net: 10/100 Ethernet
- ROM: SanDisk Ultra® microSD UHS-I 16GB

High Computing Power (HP)
- CPU: Intel Core i7 5960X (8x HT) @ 3.00GHz
- RAM: GSKill 32GB DDR4 (3200 MHz)
- Net: Intel Gigabit LAN
- ROM: Crucial MX200 500GB
Performance (I)

- **Number of instruments**

![Graph showing collection time (s) for Local DB and Remote DB for different numbers of instruments.](image)

António Dias (amcbd89@gmail.com)  
DAQBroker - CHEP 2018  
12 July 2018
Performance (II)

- Rate of data generation

![Graphs showing rate of data generation with different delays.]
Charting data request time
Under way

- **Statistics analysis**
  - comparison
  - event detection
  - forecasting

![Graph showing concentration over time with an event indicator]
Under way

- Statistics analysis
  - comparison
  - event detection
  - forecasting
- Instrument “sekeletons”
  - allow easy transfer of instruments,
Under way

- Statistics analysis
  - comparison
  - event detection
  - forecasting
- Instrument “sekeletons”
  - allow easy transfer of instruments,
- Proper installers and interfaces for binaries,
  - install as a “service” (Windows),
  - insert interfaces to control the applications
Conclusion

- DAQBroker - instrument monitoring framework
  - Automated storage, management and display of instrument data
  - Integrate in existing infrastructure (OS, DB independent)
  - Python modules (DAQBrokerServer, DAQBrokerClient)
- Ideal for flexible long term monitoring
  - Slow to medium data generation speed
  - Continuously changing instrument environments
- More information
  - Main : daqbroker.com
  - Documentation : http://daqbroker.com/documentation/
  - Github : https://github.com/daqbroker/daqbroker

Thank you for your time!

Questions?

António Dias (amcbd89@gmail.com)