PermaSense Data Management

System documentation and tutorial for online data access

April 9, 2014

Samuel Weber, Jan Beutel, Christoph Walser
THE PERMASENSE SYSTEM

This section gives an overview about:

• The PermaSense system architecture
• GSN-based data management:
  - data flow
  - servers and virtual sensors
  - web user interface
  - timing information
PermaSense System Architecture

Sensor Node (SN)
• Collects data from different sensor options

Wireless Sensor Network
• Forwards the collected data over a 868 MHz wireless communication channel to the base station

Base station
• Sinks all data and forwards it to a central server over an IP network

Backend- and GSN-Server
• Collects data from all deployments and implements a number of management and monitoring services

Global Sensor Network (GSN)
• Data streaming framework from EPFL
• Organized in “virtual sensors”, i.e. data types/semantics
• Hierarchies and concatenation of virtual sensors enable on-line processing
• Dual architecture translates data from machine representation to SI values, adds metadata

Metamodel
- Position
- Sensor type
- Validity
- Period

Rev 1.1 / page 3
Data from field site is received by the private GSN server “as is” and stored in a primary database.

- All data from sensor nodes to backend is transmitted in packets
  The length of the sensor network data packets is limited to 23 bytes. In cases where sensor data is larger, several consecutive packets are generated.

- Data is received and sorted according to data types

Data is passed on to a public GSN server where it is mapped to positions, sensor types and converted to convenient data formats.

Data is downloaded, analyzed and plotted using external tools.
Structure of a Virtual Sensor

- Data in database (GSN) is organized as virtual sensors (VS) per deployment. If there are multiple sensors yielding the same data types, this data is multiplexed into the same VS.

- Each VS has a unique name: `<deployment>_<sensor type>_<processing step>`
  Processing steps: `raw` \(\rightarrow\) `mapped` \(\rightarrow\) `converted`

- Each VS has three tabs with different information

- Each VS contains a header. The header for sensor network specific data contains:
  - `position`: number of physical locations
  - `device_id`: mapped device id at this position
  - `timed/generation_time/timestamp`: there are several time formats that are described later
  - `sensortype`: sensor types and calibration constants connected at a given position
  - `sensortype_serialid`: 
  - `header_seqnr`: sequence number denoting successively generated sensor network packets
  - `header_originatorid`: the same as device id
  - `header_atime`: packet transfer time in seconds, used to calculate generation time
  - `payload_sample_valid`: flag which specifies data acquisition errors
  - `payload_sample_no`: sample number denoting packets originating from same data sampling period (typ. period is 120 sec)
All virtual sensors are listed and sorted by deployment, data type, sensor type and process step

Manual download of data

Topology and table with network information sorted by deployment

Log files from core stations

Backend, MySQL and Zabbix monitoring server specific plots

Sensor network specific plots

Collection of scientific plots

Weather information and webcams
Global reference time (UTC) is often not available

→ Solution: Elapsed time on arrival
  • Sensor nodes measure/accumulate sojourn time
  • Base station annotates data with arrival time (e.g. UTC)
  • Generation time is calculated as difference between arrival time and sojourn time

All data carries multiple timestamps
  – *generation_time* depicts time when data is sampled
  – *timestamp* denotes the time when data reaches a UTC synchronized time base for the first time (e.g. base station)
  – *timed* is database time, i.e. the time data is inserted into the database

GSN time
  – Is in UTC (Switzerland: winter +1h (CET), summer +2h (CEST))
  – Is unix-time in millisecond!
    • *Unix time stamp* is merely the number of seconds between a particular date and the Unix Epoch (January 1, 1970).
    • *GSN time stamp* is merely the number of milliseconds between a particular date and the Unix Epoch.

Example for time conversion in Matlab:

```matlab
function t=time_gsn2matlab(st)
% converts times in gsn format to time in matlab datenum
% format, UTC time GMT+2
st=datenum([1970,1,1,0]);%Unix start time
st = st + datenum((t(1)*3600+60*t(2)+t(3))/86400);%convert to UTC
st = st + datenum((t(4)*3600+60*t(5)+t(6))/86400);%convert to GMT+2
```

\[ \hat{t}_b = t_b - \tilde{t}_b = 2013/06/06 17:47:11 - (4+1+2) = 2013/06/06 17:47:04 \]
**Metadata for Deployments**

- PermaSense maintains four deployments, Matterhorn (MH), Jungfraujoch (JJ), Dirruhorn (DH), Aiguille du Midi (ADM)
- Metadata for each deployment is described in the *nodeposition.xls* file
  - **Name** of the deployment
  - **Position**: physical location of a sensor type (shown in an additional map or overview picture)
  - **Sensor type** used at the position
  - **Coordinates** of the position
  - **Device ID**: unique ID identifying a piece of hardware

If an additional sensor gets installed, a new position is created
- If a sensor gets replaced, a new data column is added with the new device ID

<table>
<thead>
<tr>
<th>Position</th>
<th>Label</th>
<th>Name</th>
<th>Comment</th>
<th>Host Name</th>
<th>Sensor ID</th>
<th>Sensor Type</th>
<th>CR Length</th>
<th>GPS X</th>
<th>GPS Y</th>
<th>GPS Alt</th>
<th>Date</th>
<th># Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Base Station</td>
<td>permasense-caa-adm-1</td>
<td></td>
<td>NCTN</td>
<td>100</td>
<td>6.887489</td>
<td>45.879446</td>
<td>3750</td>
<td>2021</td>
<td>2021</td>
<td>31.05.2013</td>
</tr>
<tr>
<td>1</td>
<td>West Face Piton Central Top Crack</td>
<td>1</td>
<td>NCTN</td>
<td>100</td>
<td>6.88762</td>
<td>45.87993</td>
<td>3750</td>
<td>2121</td>
<td>2121</td>
<td>2121</td>
<td>06.05.2013</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>West Face Piton Central Bottom Crack</td>
<td>2</td>
<td>NONN</td>
<td>150</td>
<td>6.887391</td>
<td>45.879083</td>
<td>3750</td>
<td>2105</td>
<td>2105</td>
<td>2105</td>
<td>08.07.2012</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>West Face Piton Nord</td>
<td>3</td>
<td>NONN</td>
<td>200</td>
<td>6.886823</td>
<td>45.879243</td>
<td>3750</td>
<td>2110</td>
<td>2110</td>
<td>2110</td>
<td>08.07.2012</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>South Face Piton Nord</td>
<td>4</td>
<td>NCTN</td>
<td>150</td>
<td>6.886902</td>
<td>45.879011</td>
<td>3750</td>
<td>2109</td>
<td>2109</td>
<td>2109</td>
<td>08.07.2012</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>East Face Piton Nord</td>
<td>5</td>
<td>NONN</td>
<td>100</td>
<td>6.887519</td>
<td>45.87931</td>
<td>3750</td>
<td>2106</td>
<td>2106</td>
<td>2106</td>
<td>08.07.2012</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>North Face Piton Nord</td>
<td>6</td>
<td>NCTN</td>
<td>200</td>
<td>6.887177</td>
<td>45.879532</td>
<td>3750</td>
<td>2111</td>
<td>2111</td>
<td>2111</td>
<td>08.07.2012</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>Borehole South</td>
<td>7</td>
<td>CR1000</td>
<td>8.886813</td>
<td>45.87913</td>
<td>3750</td>
<td>2120</td>
<td>2120</td>
<td>2120</td>
<td>08.07.2012</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Borehole North</td>
<td>8</td>
<td>CR1000</td>
<td>8.886813</td>
<td>45.87913</td>
<td>3750</td>
<td>2120</td>
<td>2120</td>
<td>2120</td>
<td>08.07.2012</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Borehole East</td>
<td>9</td>
<td>CR1000</td>
<td>8.886813</td>
<td>45.87913</td>
<td>3750</td>
<td>2120</td>
<td>2120</td>
<td>2120</td>
<td>08.07.2012</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Relay South Piton Central</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Argentière Glacier Webcam</td>
<td>11</td>
<td>Webcam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Aiguille du Midi Webcam</td>
<td>12</td>
<td>Webcam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Aiguille du Midi Webcam</td>
<td>13</td>
<td>Webcam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Mont Blanc Webcam</td>
<td>14</td>
<td>Webcam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Relay North Top Piton Central</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Relay West Terrace Piton Nord</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Relay Cable Car</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DATA ACCESS

This sections gives an overview about data access using external tools.
Accessing Data from the GSN Server

- There are two different methods to get data from GSN using **http queries**:
  1. **One-shot query approach**: each request returns data based on the database state at the time of the query. This allows one to quickly obtain aggregated data from a virtual sensor and export this data in convenient formats, e.g. CSV or XML.
  2. **Streaming approach**: requested data is continuously streamed to the user in real-time until the connection to the server is closed.

- Example for a simple *one-shot query* without aggregation/conditions
  - Suppose, you want to query all fields of the *matterhorn_crackmeter__tctc* virtual sensor (2 crackmeters, 2 thermistors) between 25/08/2012 and 13/06/2013 (UTC):
    1. Open in browser:
       ```
       http://data.permasense.ch/multidata?vs[0]=matterhorn_crackmeter__tctc&time_format=iso&field[0]=All&from=25/08/2012+00:00:00&to=13/06/2013+00:00:00
       ```
       where:
       - **vs[0]**: name of virtual sensor
       - **time_format**: time format of returned data
       - **field[0]**: list of data fields to return
       - **from**, **to**: time limits of data request in UTC

       For further options and syntax information please consult:
       ```
       https://github.com/LSIR/gsn/wiki/Web-Interface#multidata
       ```
    2. You will get a CSV-formatted file with the requested data.
  - A complete example using Matlab can be downloaded

Rev 1.1 / page 10
Accessing Data over the *data.permasense.ch* Web Interface

**http://data.permasense.ch**

- Manual access to all virtual sensors and data fields over web interface
- Example (same as for **one-shot query**):
  - Get all fields of the *matterhorn_crackmeter__tctc* VS between 25/08/2012 and 13/06/2013:
    1. Go to [http://data.permasense.ch](http://data.permasense.ch), open tab ‘DATA’
    2. In sub-tab ‘Data Output’, select ‘All Data’ from the first drop down menu and the according virtual sensor:
      ```
      Data Output
      All Data [ ] 10 Values
      Aggregation [ ] No Aggregation [ ] 2 [ ] Hours [ ]
      Timeline [ ] generation_time [ ]
      Add Output
      matterhorn_crackmeter__tctc [ ] All Fields [ ] Remove
      ```
    3. Click on sub-tab ‘Conditions’ and set the time limits:
      ```
      Conditions
      From [UTC] 25/08/2012 00:00:00 To [UTC] 13/06/2013 00:00:00
      Add Condition
      matterhorn_crackmeter__tctc [ ] All Fields [ ] Between -inf and +inf Remove
      ```
    4. Click on sub-tab ‘Results’ and access the data in one of the offered formats:
      ```
      Results
      Download [ ] Plot [ ] Table
      Download CSV [ ] Download XML [ ] Generate PDF [ ] Time Format [ ] UNIX [ ]
      ```

Note: The plot is interactive and can be used to easily preview the resulting data.
Accessing Data over the `data.permasense.ch` Web Interface

- A selection of predefined plots can be found in the tabs **SCIENCE** and **SENSOR NETWORK**

- Plots are generated using the **Vizzly** framework

- All plots are interactive:
  - **Time**: time range to be shown
  - **Pos**: data fields to be plotted
  - **Update**: has to be clicked after ticking
  - **Download CSV**: download aggregated of currently shown plot
BACKGROUND ON GSN AND SYSTEM ARCHITECTURE

This sections gives an overview about:
Vizzly: Visualization of Large Data

- Fast access to millions of data samples
- Pan, zoom, channel selection
- Combination of historic and real-time data

[Image showing a graph of Dozer Temperature with data points and timelines from January 2009 to 17/07/11, and selected time range from 06:00 to 18:00.

[Keller IPSN 2011, SenseApp 2012]
WSN On-Node Storage Layer

• On-node flash based storage (SD-Card)
  – Integrated with Dozer queuing mechanism (beacon traces & per-link ack’s with backpressure)
  – All generated packets are stored on local flash memory
  – Packets not yet sent are flagged for sending later
  – Bulk access optimized for flash memory (no single packet transfers)

• Enables both delayed sending (disruptions) and post-deployment validation
Mitigating Post WSN Data Loss

- **BackLog = Auxiliary data aggregation layer at device level**
  - Remote storage and synchronization layer for Linux systems
  - Python based, designed for PermaSense CoreStation
  - Plugin architecture for extension to custom data sources
  - Data multiplex from plugin to GSN wrapper over one socket

- **Reliable (flow controlled) synchronization**
- **Schedulable plugin/script execution, remote controlled**
Metadata Mapping Architecture

• Based on 2 GSN instances
  – Separation of load/concern across two machines
  – “Private” GSN instance, raw data, protected, high availability
  – “Public” GSN instance, mapped and converted data, open, non-critical

• Metadata stored in version control system (CSV, SVN)

• Mapping of
  – Positions, coordinates, sensor types, conversion functions, sensor calibration...

• Conversion of
  – Time formats, raw to SI values...

• Replay of metadata/mapping possible, e.g. on errors

• Change management
Metadata Change Management

- Allows simple exchange of sensor hard-/software at runtime
- Post-deployment annotation
  - Stop GSN – deployment change – annotate metadata – restart GSN
- Automatic synchronization with 1 day change boundaries