gempa

GLOBAL EARTHQUAKE MONITORING PROCESSING ANALYSIS

Web Interface

CAPS ships with a read-only Web interface which provides server statistics and which lists available data streams.

The Overview shows (figure 4) provides traffic data such as current upload rate, received bytes and packages. On the right side the same traffic data is displayed as live charts allowing to instantly recognize changes in the data flow.

The Stream perspective shows all data streams including the time window of available data. The list may be filtered to easily find the streams of interest. Also time series data as well as image data may be instantly viewed by clicking on a particular stream entry (figure 5).

Configuration

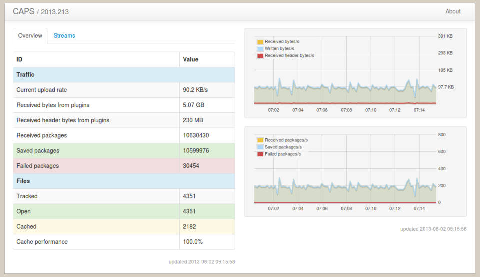
CAPS is developed as a standard SeisComP3 application. It uses the SeisComP3 infrastructure for startup and configuration. A GUI application simplifies the setup and configuration tasks. In addition gempa offers a full-feature Web-based configuration interface allowin you to control your installation from anywhere using any device running a Web browser.

Plug-ins

Since CAPS was brought to the market more and more plug-ins have been developed. The most import once are:

- **caps2caps** – Mirrors data between different CAPS instances.
- **slink2caps** – Reuse of SeedLink plug-ins, e.g. Chain, NAQS, WIN, SCREAM
- **rs2caps** – Collects data from a SeisComP3 RecordStream. This plug-in is very flexible since various data sources are already available through the SeisComP3 RecordStream interface.
- **rtpd2caps** – Fetches data from a RTPD server (REF TEK).
- **ntrip2caps** – Receives GPS data from a NTRIP Caster.
- **v4l2caps** – Connects an image or video device, e.g. a Webcam.

Web Interface

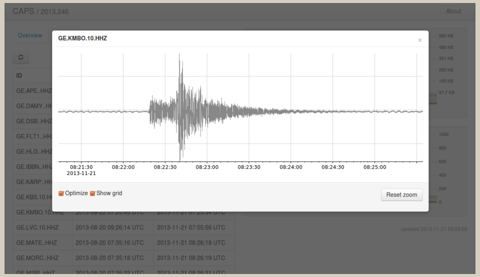

CAPS / 2013.213

ID	Value
Traffic	
Current upload rate	90.2 KB/s
Received bytes from plugins	5.07 GB
Received header bytes from plugins	230 MB
Received packages	10630430
Saved packages	10599976
Failed packages	30454
Files	
Tracked	4351
Open	4351
Cached	2182
Cache performance	100.0%

updated 2013-08-02 09:15:58

updated 2013-08-02 09:15:58

Figure 4 Overview perspective showing traffic and file statistics


CAPS / 2013.246

GE.KMBO.10.HHZ

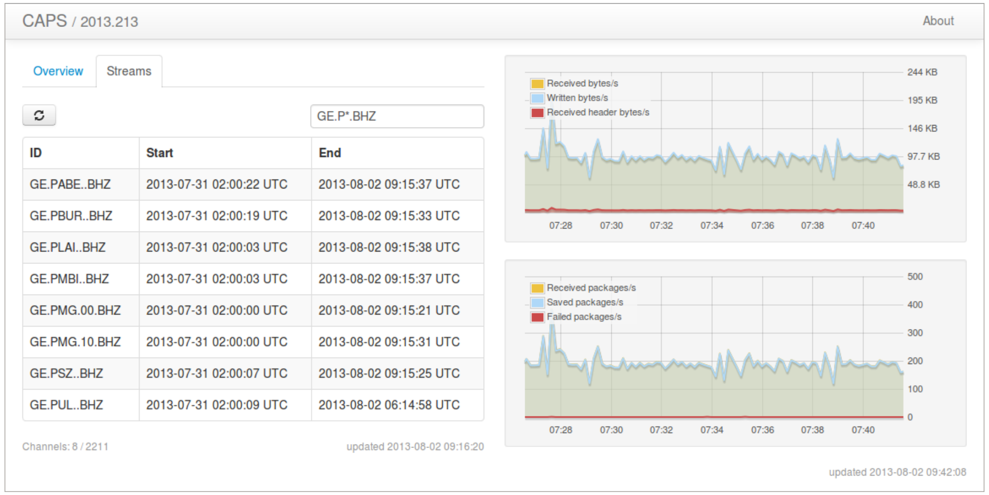
ID		
GE.LVC.10.HHZ	2013-08-20 09:26:14 UTC	2013-11-21 07:55:56 UTC
GE.MATE..HHZ	2013-08-20 07:35:16 UTC	2013-11-21 08:26:18 UTC
GE.MORC..HHZ	2013-08-20 07:35:18 UTC	2013-11-21 08:26:19 UTC

updated 2013-11-21 09:29:08

Figure 5 Stream perspective allowing to filter available streams and to view data

CAPS

Common Acquisition Protocol Server by gempa GmbH


CAPS / 2013.213

GE.P*.BHZ

ID	Start	End
GE.PABE..BHZ	2013-07-31 02:00:22 UTC	2013-08-02 09:15:37 UTC
GE.PBUR..BHZ	2013-07-31 02:00:19 UTC	2013-08-02 09:15:33 UTC
GE.PLAI..BHZ	2013-07-31 02:00:03 UTC	2013-08-02 09:15:38 UTC
GE.PMBI..BHZ	2013-07-31 02:00:03 UTC	2013-08-02 09:15:37 UTC
GE.PMG.00.BHZ	2013-07-31 02:00:00 UTC	2013-08-02 09:15:21 UTC
GE.PMG.10.BHZ	2013-07-31 02:00:00 UTC	2013-08-02 09:15:31 UTC
GE.PSZ..BHZ	2013-07-31 02:00:07 UTC	2013-08-02 09:15:25 UTC
GE.PUL..BHZ	2013-07-31 02:00:09 UTC	2013-08-02 06:14:58 UTC

Channels: 8 / 2211

updated 2013-08-02 09:16:20

updated 2013-08-02 09:42:08

Application

- real-time data transmission from remote sensor to data center
- center to center data transfer

Features

- multi-sensor data transfer, e.g. waveform data, Webcam images
- redundant data transmission
- ensures timeliness of data
- on-the-fly reconfiguration
- lightweight protocol for minimized packet overhead
- archived and real-time data served through one protocol and one connection
- reliable data transfer, retransmission of data in case of network outage or server restart
- Web interface for server monitoring and data review
- secure communication via SSL
- fine-grained access control
- connectivity to the SeisComP3 framework

gempa GmbH · http://www.gempa.de · info@gempa.de

Introduction

The Common Acquisition Protocol Server (CAPS) was developed to fulfill the needs to transfer multi-sensor data from the station to the data center. As nowadays more and more stations with co-located sensors like broadband seismometer, accelerometer, CGPS, temperature, video cameras, etc. are build up, a acquisition protocol is required, which can efficiently handle low- and high-sampled data through one unified protocol.

Timeliness of Data

The focus on early warning in the last decade gave the timeliness of data a high priority. This made it necessary to allow backfilling of data (send most recent data first) as well as reducing the size of the data records. These requirements for data acquisition can not be fulfilled by many standard software packages. For example SeedLink, the de facto standard of real-time data transmission widely used to transmit seismic data, can neither handle backfilled data nor supports a variable record size. CAPS fills this gaps as it is format independent, which allows to handle low and high sampled data in parallel and uses index files to keep track of the timely order of packets.

Redundancy

In the last years one topic became a focal point of interest - redundancy. CAPS allows to fetch data packets through several connections in parallel, for example through a satellite line and Internet. Once the data is transmitted to CAPS, is is merged with the already available data set where duplicates are dropped. This allows full redundancy in data transmission and the only single point of failure left, is the equipment at the station itself.

On-the-fly Reconfiguration

One of the main problem for institutes handling massive data is the downtime during reconfiguration. CAPS reduces the downtime to an absolute minimum and at the same time takes care that no gaps in the data occur. This is realized through two approaches. One is the support of wildcard requests, the other is the buffering of data within the plug-ins. Wildcard request means, that CAPS can request all data available at the source and doesn't need a specific list of stations. Once the source adds a station, the data is directly transmitted to the target without the need to reconfigure or restart CAPS.

Architecture

Figure 1 shows the architecture of CAPS. The central component is the server, which receives data from sensors or other data centers, stores it into an archive and provides it to connected clients. The connection between a data provider and CAPS is made through a plug-in.
Plug-ins are independent applications which, similar to clients, use a network socket to communicate with the server. The advantages of this loose coupling are:

- plug-ins may be developed independently and in a arbitrary programming language
- a poorly written plug-in does no crash the whole server
- plug-ins may run on different machines
- plug-ins may buffer data in case the server is temporary unavailable

Deployment

Figure 2 illustrates a possible deployment of CAPS and its plug-ins. The acquisition of data from other data centers is most likely done through a public interface reachable over the Internet. For this center-to-center communication a plug-in is typically launched on the receiving site to feed the local CAPS server.
For the direct acquisition of data from a sensor the plug-in has to run on the sensor station. This plug-in will send the data either directly to the data center or to local CAPS instance. The advantage of the second approach is:

- **better protection against data loss** – In case of a connectivity problem a local CAPS will store observations to the hard drive for later retrieval.
- **direct client access** – A client may directly receive data from the sensor station.
- **less packet overhead** – The CAPS client protocol is more lightweight than the plug-in protocol. A client packet only consists of a two byte header followed by the data.

The ability to connect different CAPS instances simplifies sharing of data. One protocol and one implementation is used for the sensor-to-center and center-to-center communication. In the same way multiple CAPS instances may be operated in one data center on different hardware to create backups, establish redundancy or balance the server load.

Modular Architecture

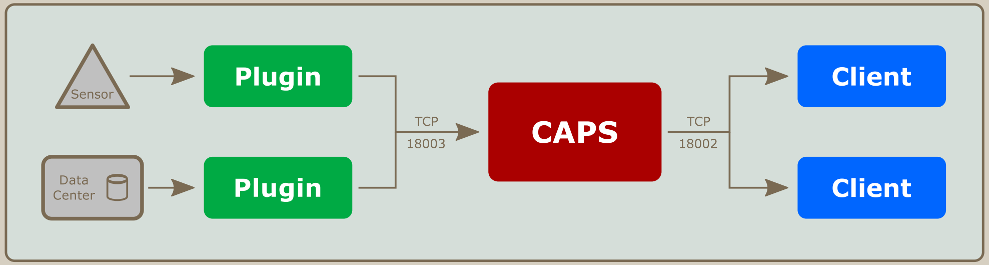


Figure 1 CAPS architecture showing the central CAPS server which is feed by acquisition plug-ins and which distributes data to connected clients.

Deployment

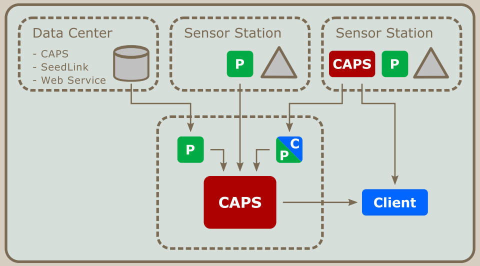


Figure 2 Deployment example showing data acquisition form centers and sensor stations

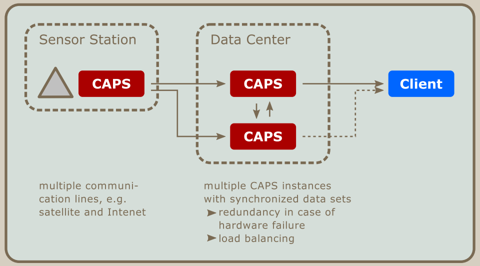


Figure 3 Redundant data transmission, storage and processing