Baikal-GVD data transfer and storage

Timur V. Elzhov (JINR)
on behalf of Baikal-GVD collaboration

VLVvT 2021

Description

Data collected by the Baikal Gigaton Volume Detector (Baikal-GVD) needs to be processed in real time at a data center located in Dubna, Moscow Oblast, approx. 4,300 km away. An infrastructure facilitating data transfer and storage was designed and implemented

In this talk, some implementation details and our approach to ensuring fault tolerance and 24/7 access to the data are presented

Baikal-GVD to JINR data transfer

Baikal-GVD to JINR data transfer

Online processing

One of our main aims is to process incoming data files as soon as possible, online in other words

The incoming data stream is split into files of 250MB. Nowadays, a new file is processed after it has been completed, but we are developing a technique where a new file being collected will also be processed at the same time

Monitoring file system changes

There are two approaches for monitoring file system changes in general:

- Polling for a new data with cron. The polling items are added to a cron table (‘crontab’)
- Subscription to changes reported by inotify Linux kernel subsystem. Subscription items can be added to a similar table (‘incrontab’) handled by incron service based on inotify (referred to as incron/inotify hereinafter)

Monitoring file system changes

In both methods, a user has to prevent clashes between successive calls (five stars in crontab means the command is called every minute)

Cron:

```
* * * * * $HOME/bin/my-command arg1 arg2
```

Incron:

```
$HOME/incoming IN_CREATE $HOME/bin/my-command arg1 arg2
```

Locking:

```
/tmp/my-command-arg1-arg2.lock
```

Monitoring file system changes (cron)

The simpler approach: using the cron service

- Pros: Quite straightforward
- Cons: Possible waste of performance. No instant reactivity – delay up to minute

Monitoring file system changes (incron)

The right approach: using incron/inotify which is designed exactly for this purpose

- Pros: True reactivity (notified about changes instantly)
- Cons: Requires a bit more care – an observing command writing something to its observed folder may get lost in an infinite loop

Monitoring file system changes (active)

Currently we are using the cron polling method, but ready to switch to the true reactive incron/inotify model

Baikal-GVD area

Baikal-GVD area

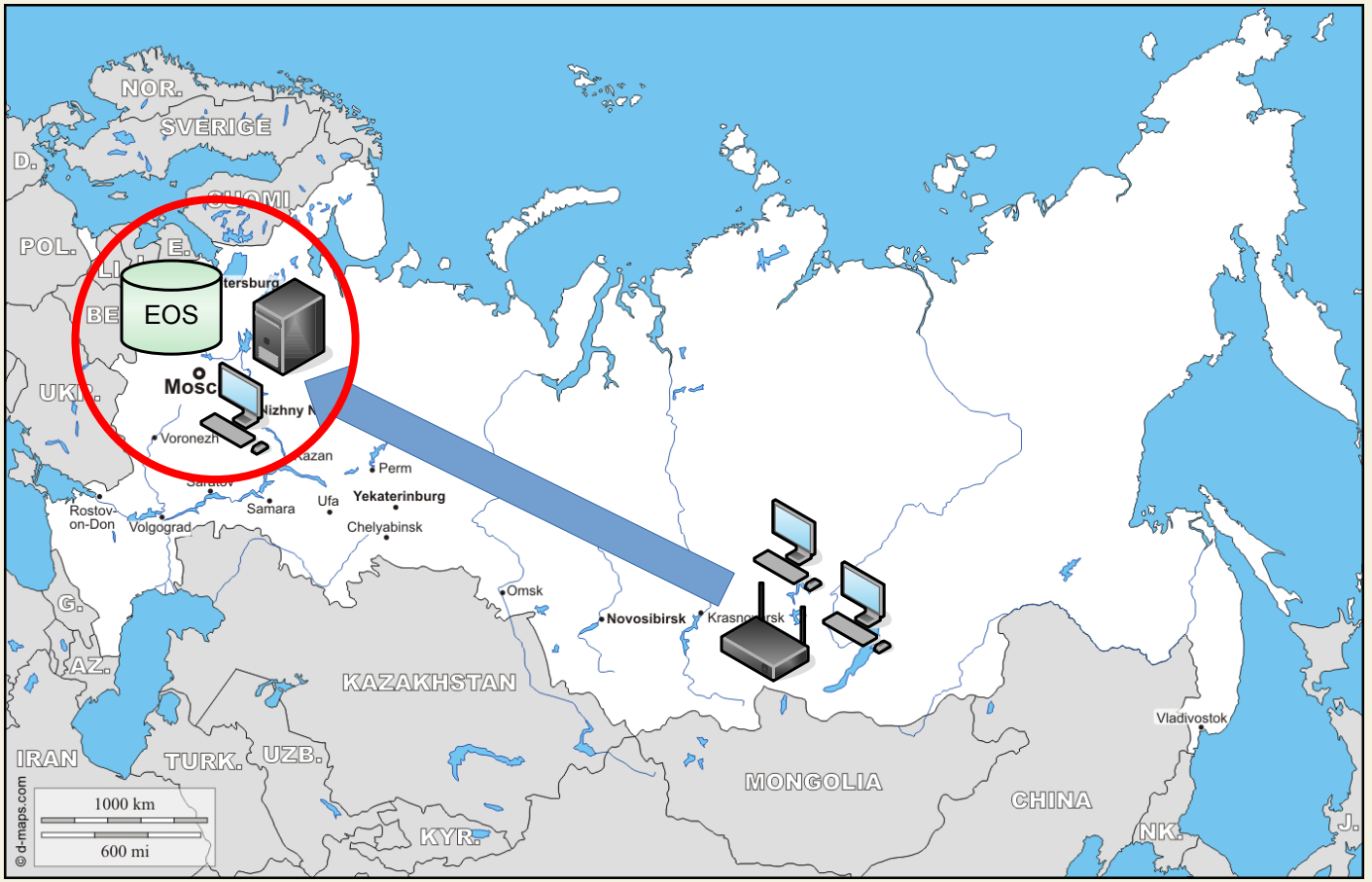
- Operating systems in use are Linux and FreeBSD
- Compressing and relocation are scheduled using cron

```
* * * * * $HOME/bin/compress-raw-files 2021
* * * * * $HOME/bin/move-compressed-raw-files 2021
```

- To prevent clashes between successive calls, the following path names are chosen:

```
/tmp/compress-raw-files-2021.lock
/tmp/move-compressed-raw-files-2021.lock
```

JINR area


NOR.
SVERIGE
D.
POL.
BE
UK
EOS
Mosc
Voronezh
Perm
Rostov-on-Don
Volgograd
Samara
Ufa
Yekaterinburg
Chelyabinsk
Omsk
Novosibirsk
G.
AZ.
KAZAKHSTAN
IRAN
TURK.
UZB.
KYR.
MONGOLIA
CHINA
Vladivostok
NK.
J.
1000 km
600 mi
© d-maps.com

JINR area

- The final destination is the JINR EOS storage
- However, the data is not transferred directly to EOS due to some EOS FUSE limitations. Instead, it's at first rsync'ed to a buffer machine, which in turn copies data to EOS by the special EOS copying tool

  The copying scheduled by the cron service as well

CERN EOS

https://eos.cern.ch/

EOS authenticates by:

- X.509 certificates (CERN grid access)
- Kerberos (local access)

Kerberos authentication in JINR

- Works on the basis of tickets
- An initial ticket (TGT) allows one to login across the Kerberos Realm's hosts without password until it has been expired
- A ticket lifetime can be extended on demand, so the real lifetime of task's permissions could be extended up to 100 days (the JINR server's limit), which is pretty enough for any GVD data processing task

Summary

- Two approaches for online processing are considered (based on cron and incron/inotify scheduling)
- Presented some technical details of the current cron-based approach
- Finally, presented some details about Kerberos authentication used by EOS in JINR

Thank you for attention!