Cell Management and Monitoring

Christof Hanke

Christof.hanke@rzg.mpg.de

Motivation

Goal: Provide a good service

- data-security
- good performance
- assistance in case of trouble
- proactively manage cell
  (move volumes before partition is full)

Thus, you need to know what's going on and have an idea (concept) how to run your cell (by scripts).

Motivation

After cell-setup, three timescales :

- Long-term :
  - how to backup
  - how to monitor (gather information)
- Mid-term :
  - Re-distribute volumes over servers
  - Integrate new servers
- Short-term :
  - react on events.

Data Security

- Have a backup-strategy.
  - test it (“backup works, only the restore failed...“)
  - RZG: remote RO-Volume for each RW-Volume.
  - Each fileserver releases his own RWs each night.
  - Tape-backend is for OSD-Volumes.
- Con: twice the disk space
- Pro: fast recovery

  Need to check that RAIDs/Filesystems containing the RO are ok.

Monitoring

Monitoring = gather information

- Stats about characteristic numbers :
  - IP-Traffic
  - RPC/sec
  - State/Location of Volumes, Partitions
- Events :
  - e.g. partition full

Monitoring

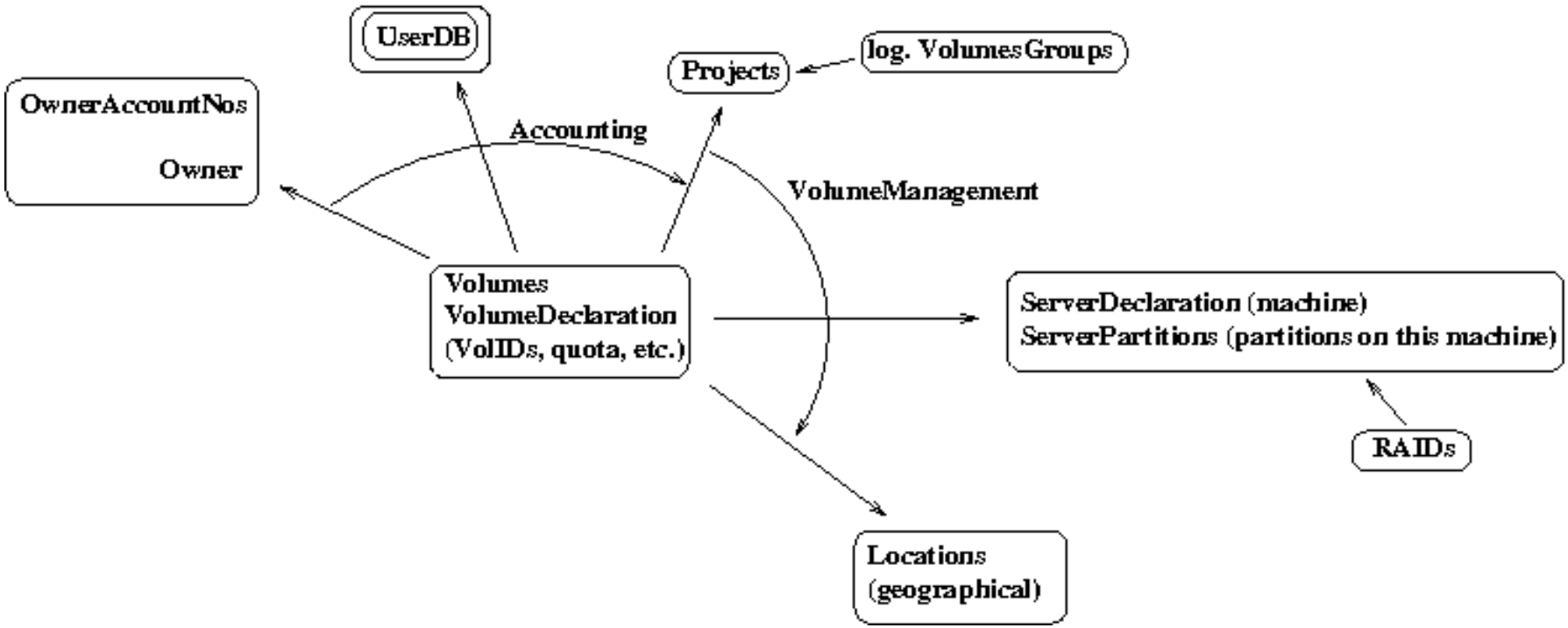
- live :
  - nagios (AFS-services, OS (e.g. NTP))
  - syslog-ng (low-level like SCSI)
  - scout (textbased and webbased)
- once a day:
  - location and state of all volumes in SQL-DB
  - web-frontend

Cell organisation

- Ext. DB to store relevant information.
- Most important: volume locations
  - geographical location
  - which server/partition
- Volumes belong to logical volumegroups (e.g. sun*)
- log. volumegroups belong to projects
  - Each project has a preferred geographical location and preferred servers for their volumes.

Cell organisation

- web-frontend prototyped in plone (zope)
- DB-Layout :



Tools

- Nagios :
  - popular monitoring system (even on ISS)
    - extended by pnp4nagios (RRD-Graphs)
    - extended by nagvis (customiseable maps)
- syslog-ng:
  - store logfiles on one loghost
  - use sth. like Simple Event Correlator for datamining
- Homebrew :
  - scripts to populate database, release volumes etc.

nagios

- Plugin based monitoring and alarming system.
- Plugins written easily
- Can be used to gather long-term statistics (pnp4nagios creates RRD-files)
- Distributed/ hierachical monitoring (one nagios in DMZ, one in internal network)
- Active and passive checks, locally and remote.

nagios

- Plugin-specifications:
  - Anything executable
  - One line of output
  - Exit code defines status
    (0 OK, 1 Warn, 2 CRIT)
  - Performance data can be included in output and then graphed with pnp4nagios

nagios-plugins

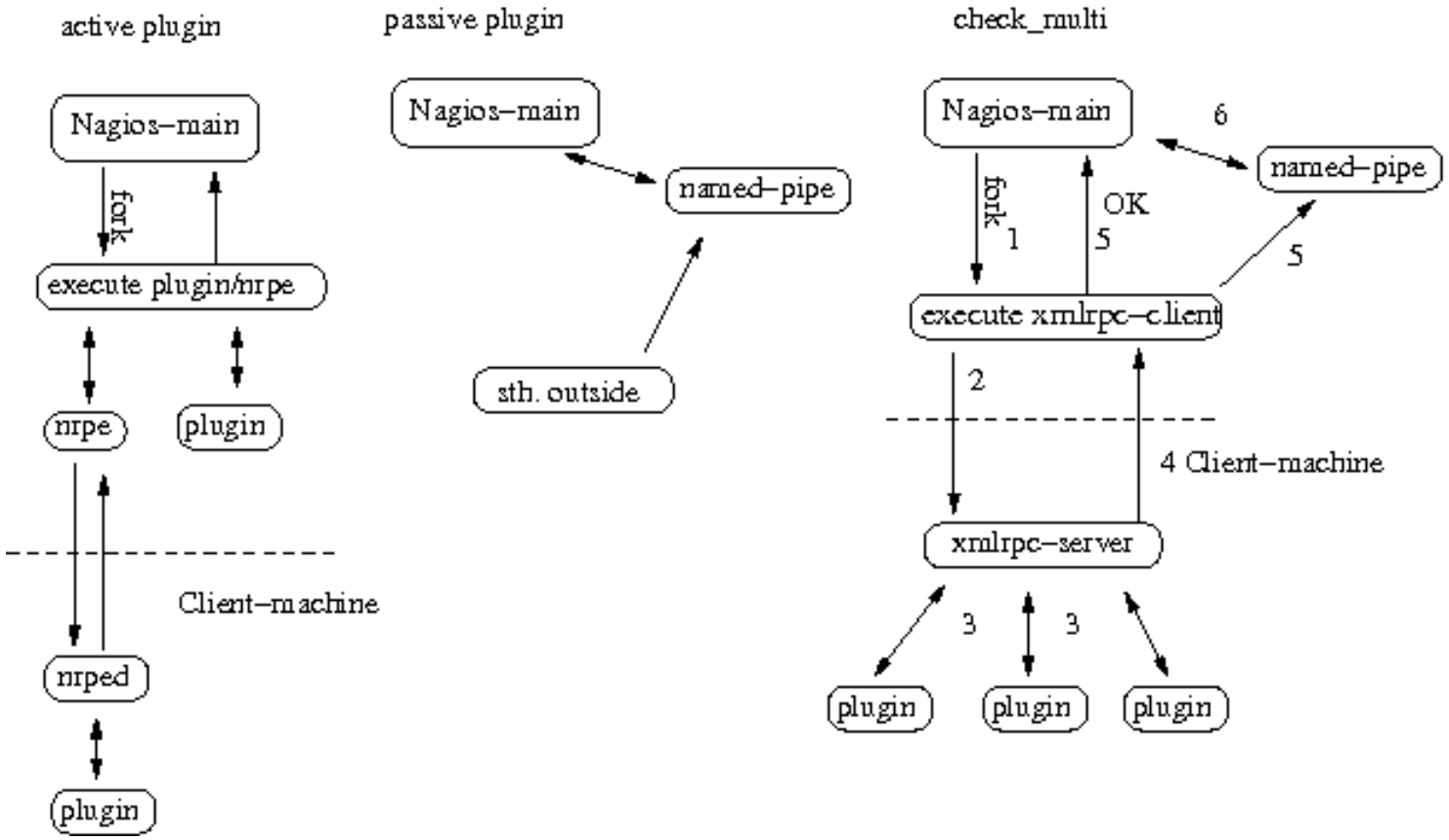
- rxdebug for fileserver,volserver, bosserver, etc.
- udebug for db-server
- KDC-replication
- Usage of partitions (OS and /vicep)
- Status of volume-release

  for stats :
- IP-traffic
- RPC-stats (not in upstream openAFS)

nagios – plugins

- Each check is a separate process → expensive.
- check_multi plugin, based on the idea of check_mk :
  - one active check injects many results as passive checks
  - can graph differences

nagios – plugins



nagios – plugins

- Each check is a separate process → expensive.
- check_multi plugin, based on the idea of check_mk :
  - one active check injects many results as passive checks
  - can graph differences

nagios- demo

- Show some stuff from RZG nagios
- Including nagvis
- Including pnp4nagios

Thank you.