High Availability Methods @ GSI

- overview on commercial methods - services and setups (Microsoft cluster methods, Oracle RAC)
- Open Source methods (linux based, individual combination)
  - tools
  - services and setups
- hardware solution
- experiences

Common HA Components

- some kind of „*heartbeat*“ between cluster nodes
- virtual „*service*“ ip address
- node configuration - active/active or active/passive
- shared data
- monitoring tools

- reliable hardware
- raid systems, SAN storage

Overview Windows Techniques

- Microsoft Clustering
- nodes communicate over second private network connection
- monitoring done by MOM (E-Mail and statistics)

(Holger Goeckel)

Windows Services and Setup

- Exchange server
  - 2 nodes
  - active/passive
  - shared SAN file system
- file server
  - 2 nodes
  - active/active (two file system partitions)
  - shared raid 5
- print server
  - 2 nodes, shared disk for printer queue, active/passive

Overview Oracle DB

central data base - „Real Application Cluster“

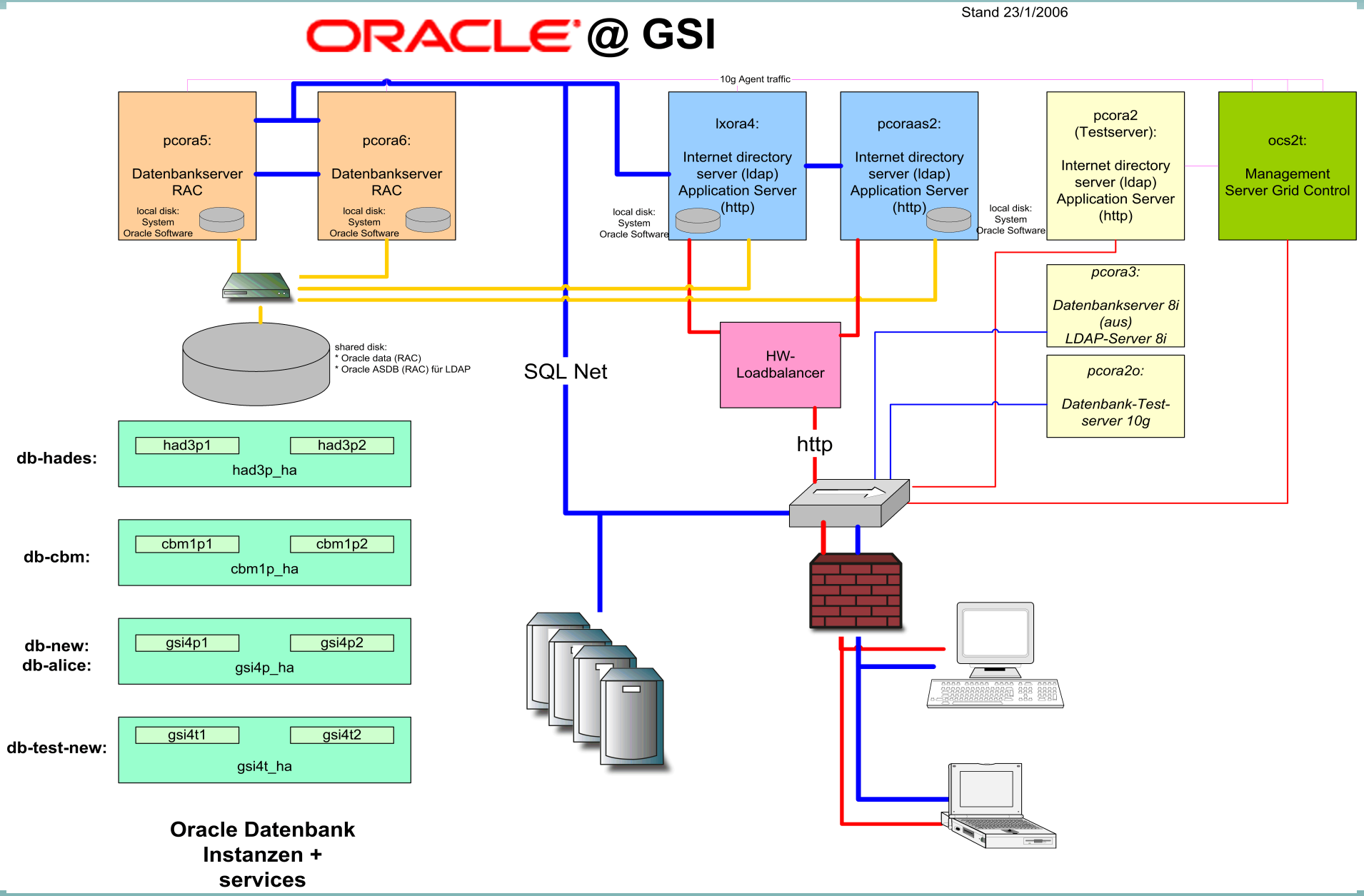
- active/active
- load balancing
- integrated monitoring (statistics, mail and sms)
- heartbeat and „Cache Fusion“:
  - every node knows transaction status of other nodes
  - on failover a running session is continued on the other node!

(Michael Dahlinger)

RAC Configuration

- two nodes with two real, two virtual and two private addresses
- shared SAN storage
- running on SuSE Enterprise Server (linux system has to be modified)
- experiences – excellent!

Oracle Real Application Cluster



Linux HA - Tools

All tools are OpenSource, GPL or similar:

- heartbeat package
  - communication between the two nodes, starts the services
- drbd
  - synchronisation of file system over network
- mon
  - system monitoring

Heartbeat Package

- does the communication between nodes via ethernet and / or serial line
- starts the HA services
- triggers a failover if communication is down
- „stonith“ - kills the other node
- checks network connectivity

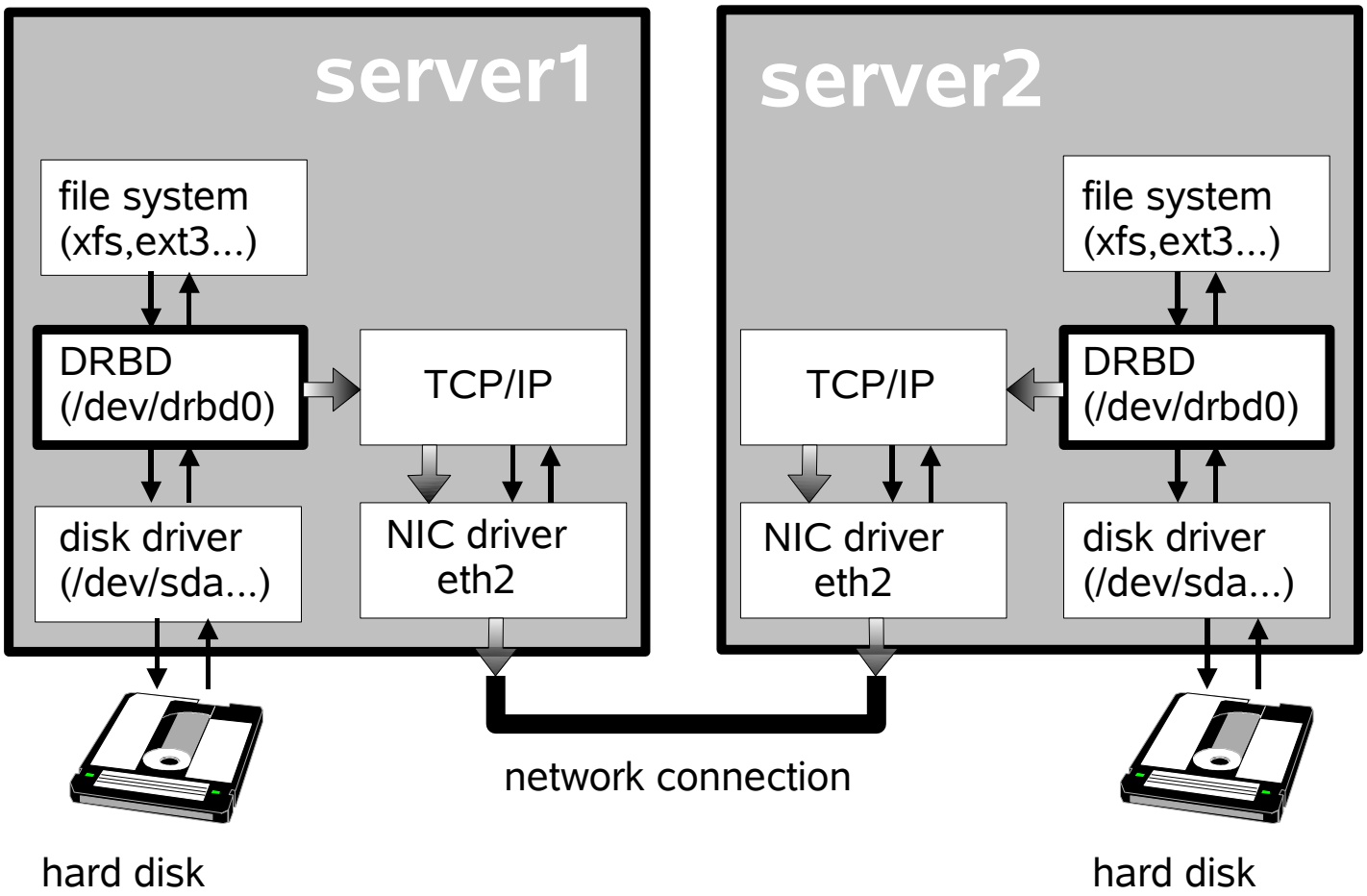
✗ heartbeat does not check if services are running!

- newer version of heartbeat has monitoring functionality integrated

DRDB

- **D**istributed **R**eplicated **B**lock **D**evice
- kernel patch which forms a layer between block device (hard disk) and file system
- over this layer the partitions are mirrored over a network connection, in principle:
  - RAID-1 over network
- each node has its own storage (raid, disk...)
- only one of the two parts can be used (primary/secondary) – changing (version 0.8)

DRBD - How it Works



(Dis-)Advantages of DRBD

- data exist twice
- real time update on slave (--> in opposite to rsync)
- consistency guaranteed by drbd: data access only on master - no load balancing
- fast recovery after failover

overhead of drbd:

- needs cpu power
- write performance is reduced (but does not affect read perfomance)

Linux Monitoring - Features

Mon – perl based service monitoring daemon:

- monitors resources, services, network, server problems...
- triggers an action (in case of failure) - email, failover, service restart, reboot...
- very flexible, can be adapted to special requirements
- service is provided by monitoring and alert scripts

Linux Monitoring - Setup

- mon on central monitoring server
  - checks if the HA services are running on the master (virtual service ip)
  - checks status of all nodes (ping...)
  - checks if mon is running on nodes
- monitoring within the cluster
  - nodes in a cluster check each other

In case of failure:

- triggers a failover, a service restart, a reboot ...and sends information messages

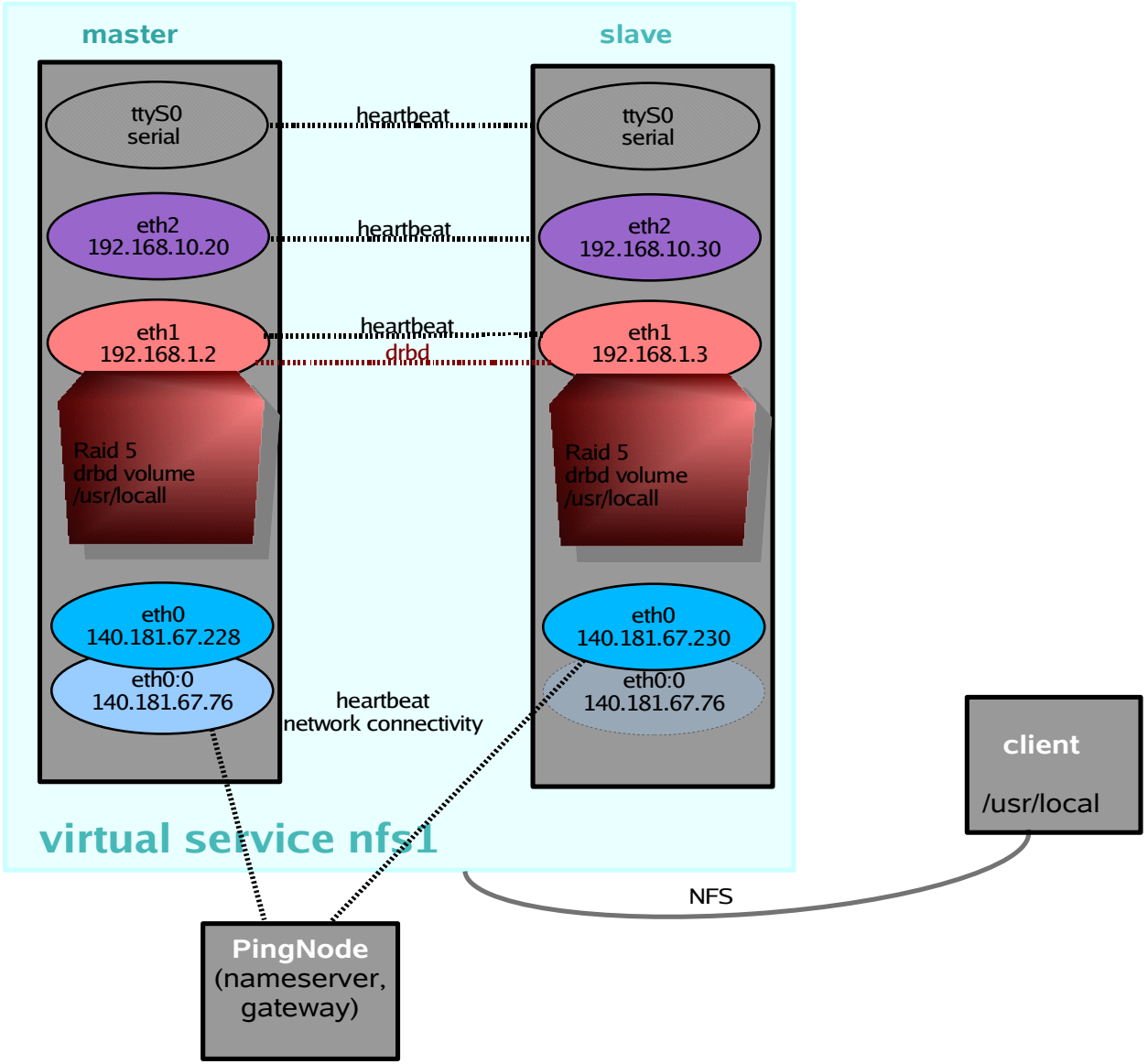
Linux HA Services

- central file server (NFS)
- entry server for storage system
- central web server
- future project - entry server for new data file system

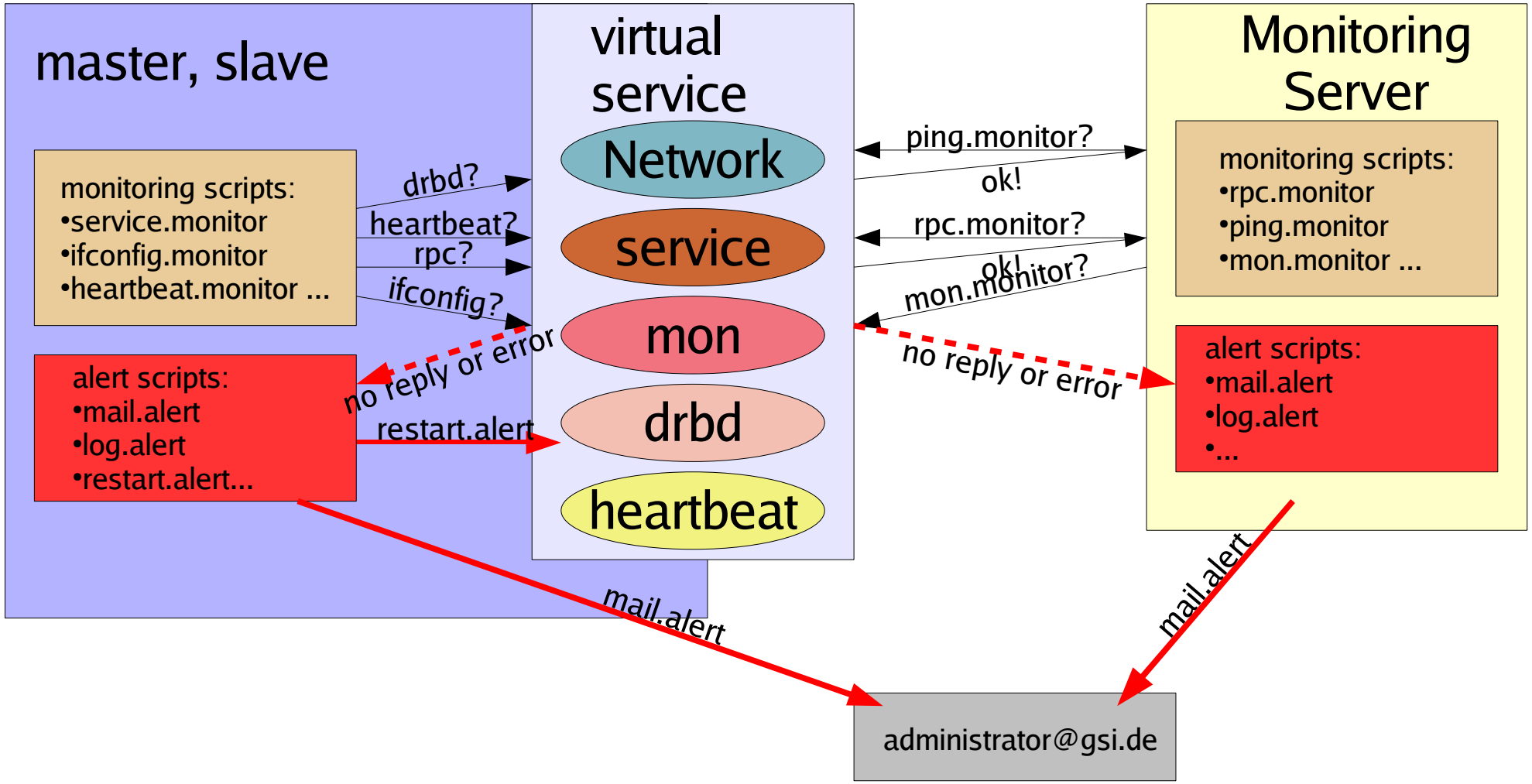
NFS Server

- central /usr/local for all linux clients
- active/passive 2 node cluster
- exported file system on drbd over raid 5
- external and internal monitoring (mon, stonith, watchdog)

Setup NFS Server



Monitoring Scheme



gStore Entry Server

- linux server pair (heartbeat)
- shared disk storage for DBs (sync via drbd)
- no access to GSI mass storage in case of failure
- gStore functions on Entry Server:
  - entry for all gStore client requests
  - read cache administration
  - write cache administration
  - file query in TSM (tape manager) and cache Dbs
  - data mover selection (load balancing)

(gStore -> Horst Goeringer)

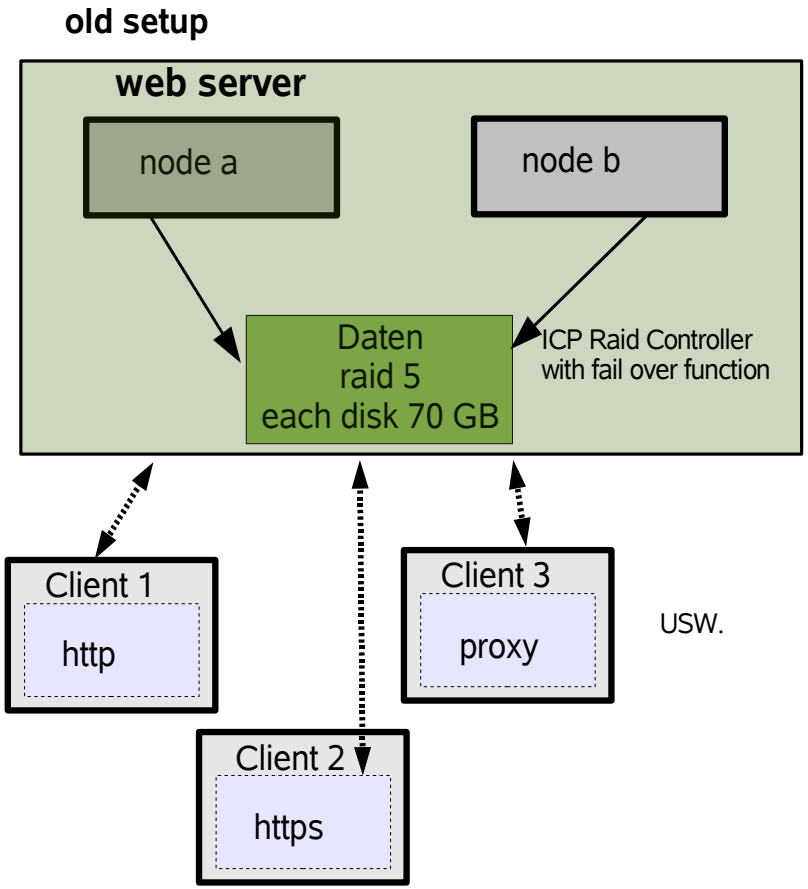
Central Web Server

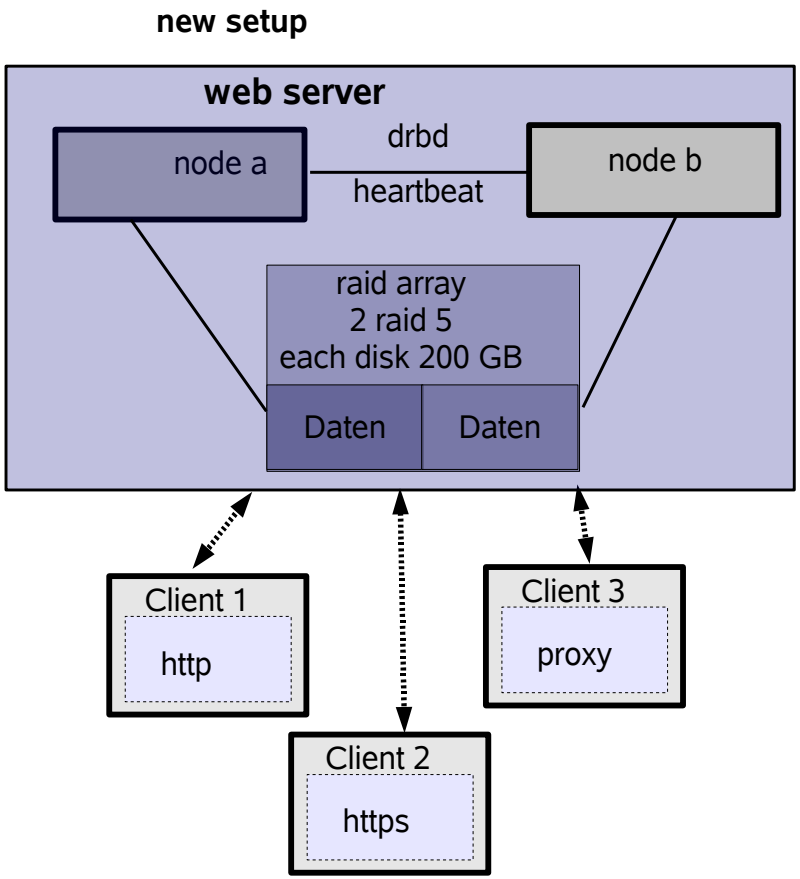
- history – ICP-Raid Controller with failover facility (module option)
- failover mode not supported by newer versions of linux (no sources available)

solution

- new disks with more space, disk space divided in 2 raid 5, connected with drbd

Web Server History





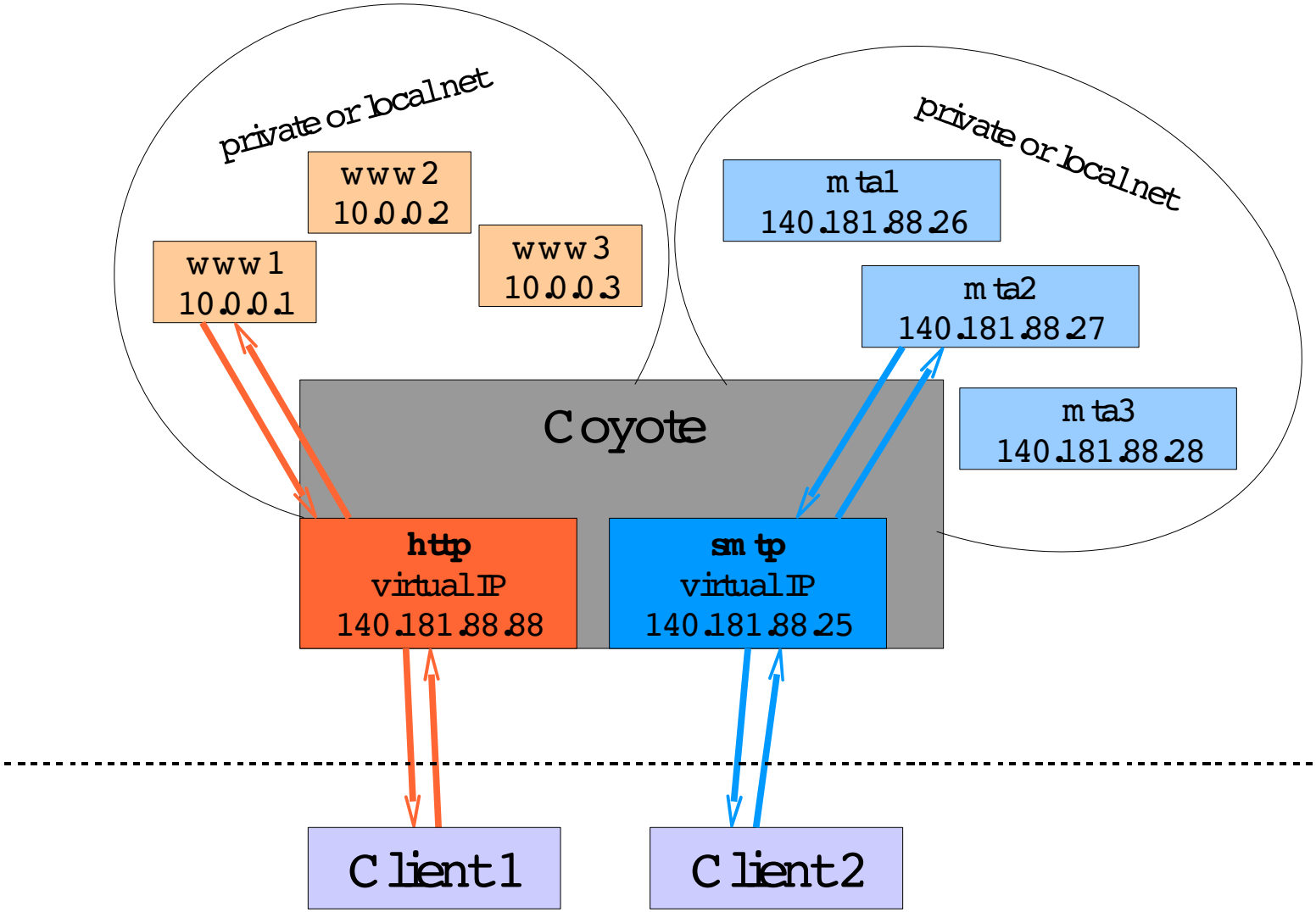
Web Server – Upgrade Problems

- no new installation, but upgrade of productive system
- critical combination of drbd version and kernel killed master node
- update to new drbd version made installation stable

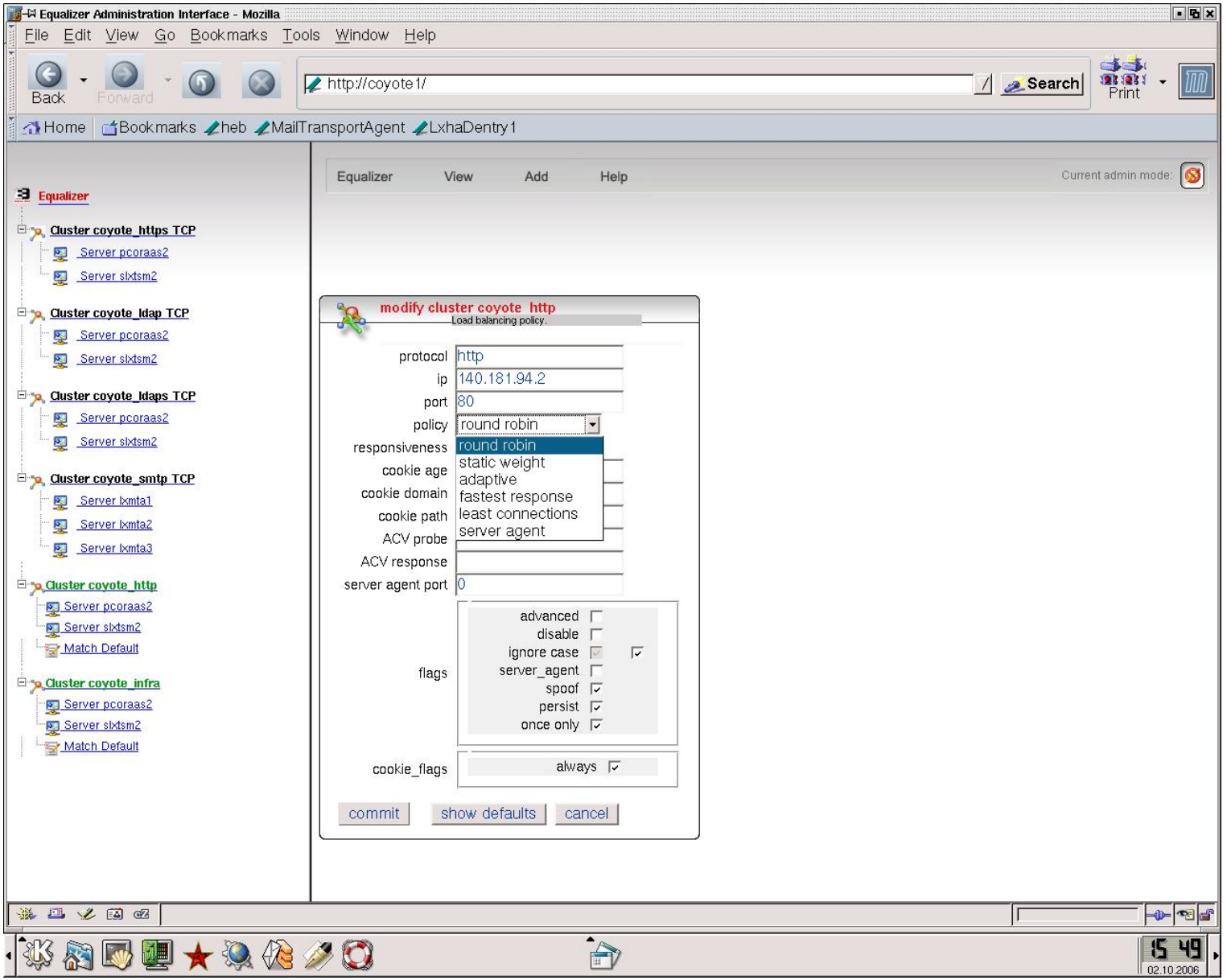
Hardware Test

- Coyote Equalizer Traffic Management
- connected by Gbit to local network
- can handle up to 65 000 virtual addresses
- fail over
- monitoring
- load balancing

Coyote Scheme



Coyote Screenshot



Experiences

- so far very good with all systems
- most failovers were related to network problems
- heartbeat/drbd/mon works very well, but installation and configuration is complicated

Future Improvements

- using coyote for Oracle application server
- using coyote instead of heartbeat for central web server and possibly more services such as DNS and MTA
- testing drbd 0.8 - write acces on both nodes - active /active cluster
- using heartbeat monitoring functionality