

Merlin

Nürnberg, 01.06.2010

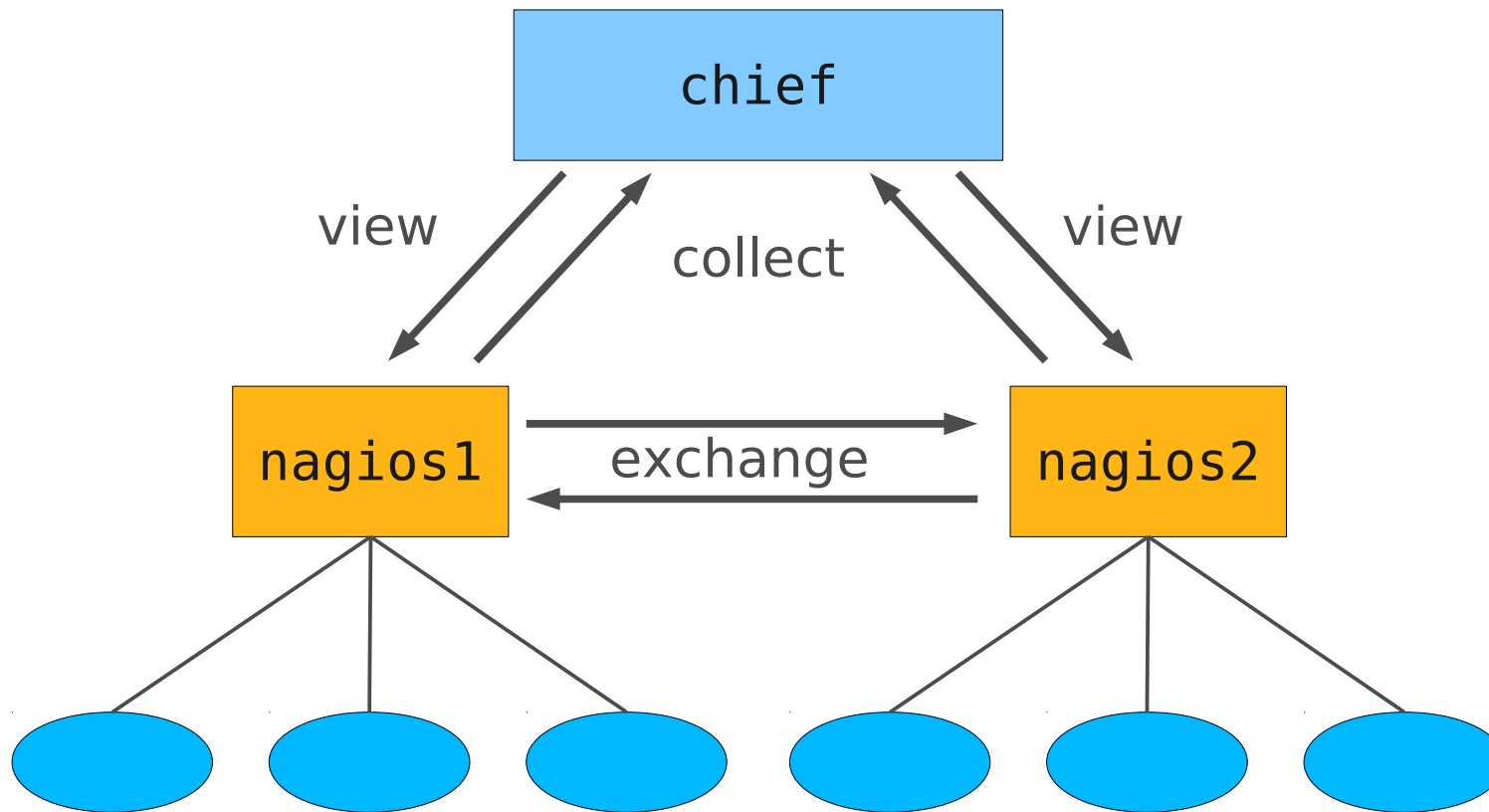
Urgeschichtliches

„Merlin war ursprünglich für
verteiltes Monitoring gedacht
– als Ersatz für NSCA –
und nicht als Alternative
zu NDOutils“

Aufgedrösel:

- Verteiltes Monitoring (a la op5)
 - ◊ redundant
 - ◊ distributed
 - ◊ redundant + distributed
- NDOutils:
 - ◊ Eventbroker-Modul
 - ◊ Daemon
 - ◊ Datenbank, Datenbank-Layout

Verteiltes Monitoring



Verteiltes Monitoring

- „view“:
 - ◊ zentrale Sicht auf verteilte Installationen
 - ◊ Thruk, Multisite, aber auch NagVis
- „collect“:
 - ◊ zentrale Instanz sammelt alle Daten
 - ◊ „obsessive compulsory service processor“, Merlin
- „exchange“:
 - ◊ zwei gleichrangige Installationen tauschen Informationen aus
 - ◊ Merlin

Redundantes Monitoring

- ... ist ein Sonderfall von verteiltem Monitoring: zwei (oder mehr) Nagios-Instanzen überwachen die gleichen Hosts/Services
- Varianten:
 - ◆ einfach: parallele Checkausführung
 - ◆ fortgeschritten: Informationsaustausch, dadurch Arbeitsteilung
 - ◆ Clusterlösungen

Verteiltes Einsammeln != verteiltes Nagios

- Eine einzelne Nagios-Installation kann ebenfalls verteilte Information einsammeln:
 - ♦ check_multi
 - ♦ check_mk
 - ♦ NSCA + Addons (z.B. Cron)
 - ♦ DNX (Distributed Nagios eXecutor)

Merlin: Status

- Focus in Branch „master“
 - ◊ Datenbank für Standalone-Host
- Full featured: Branch „next“
 - ◊ Geplantes Release: August 2010
- Die nachfolgend gezeigten verteilten Szenarien sind derzeit noch nicht (stabil) verfügbar und basieren auf den Spezifikationen von op5!

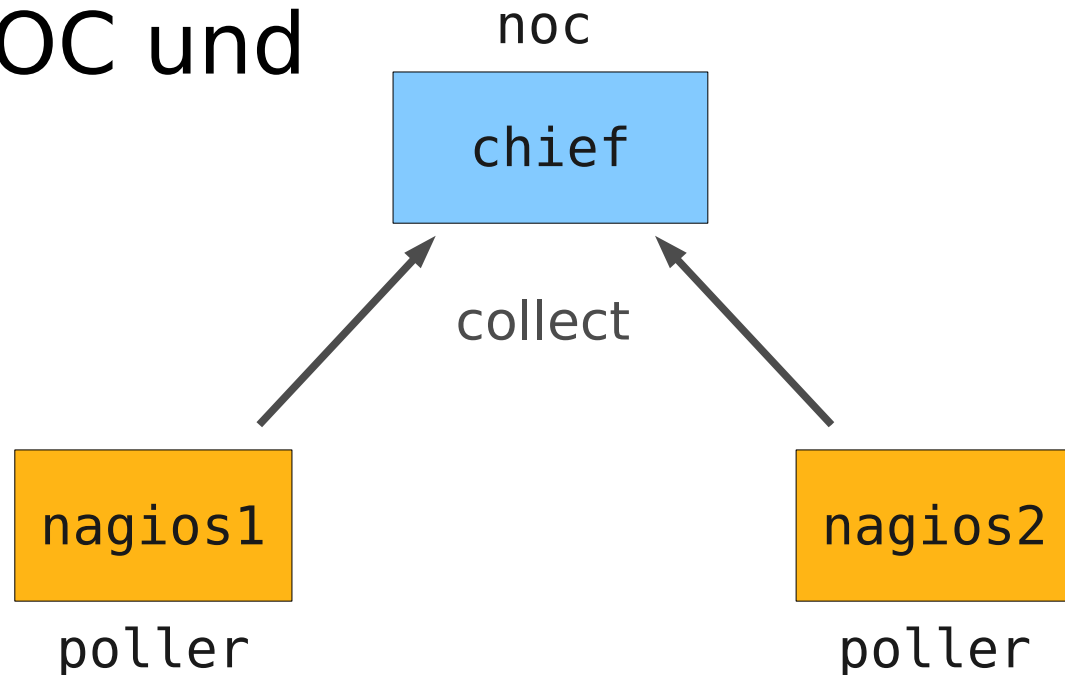
Merlin redundant

- Merlin:
(fast) alle Informationen werden an die anderen Instanzen übermittelt
- Arbeitsteilung:
den Check führt aus, wer diesen zuerst eingeplant hat.

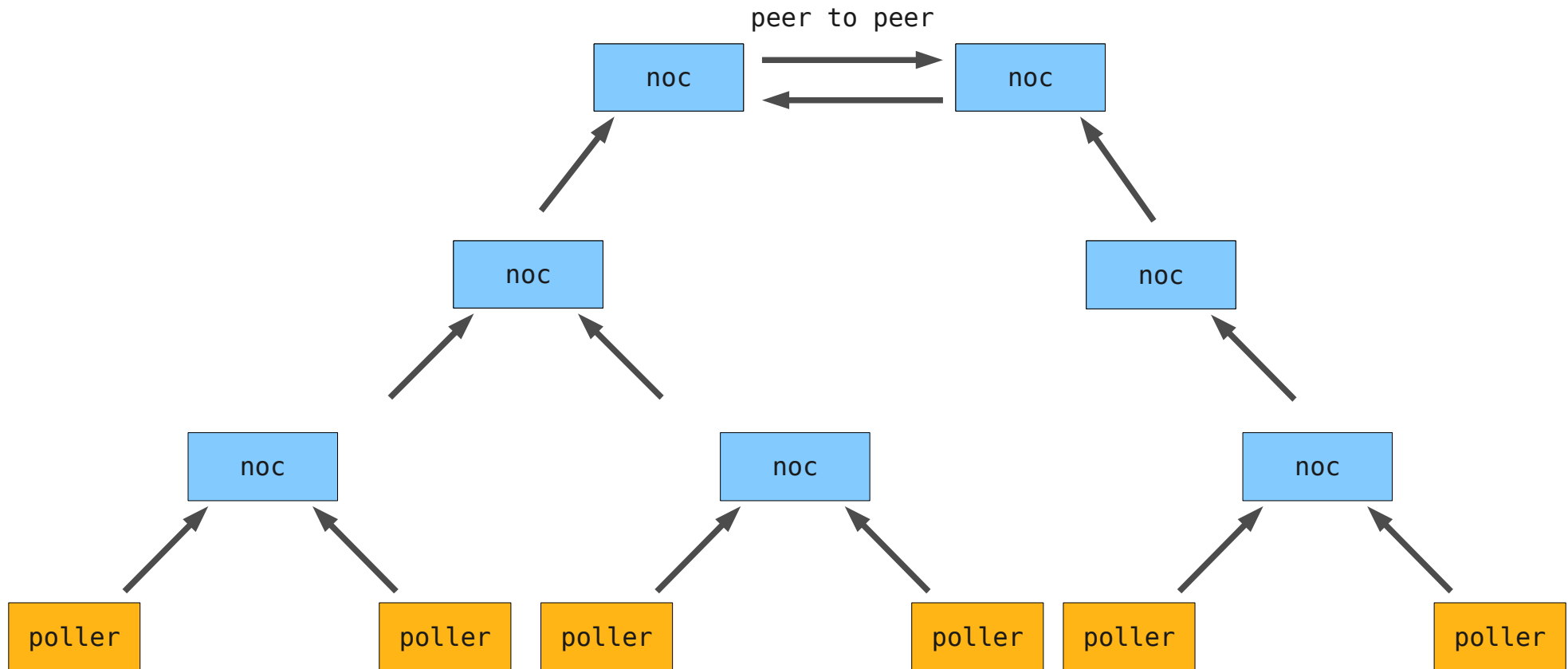


Merlin verteilt

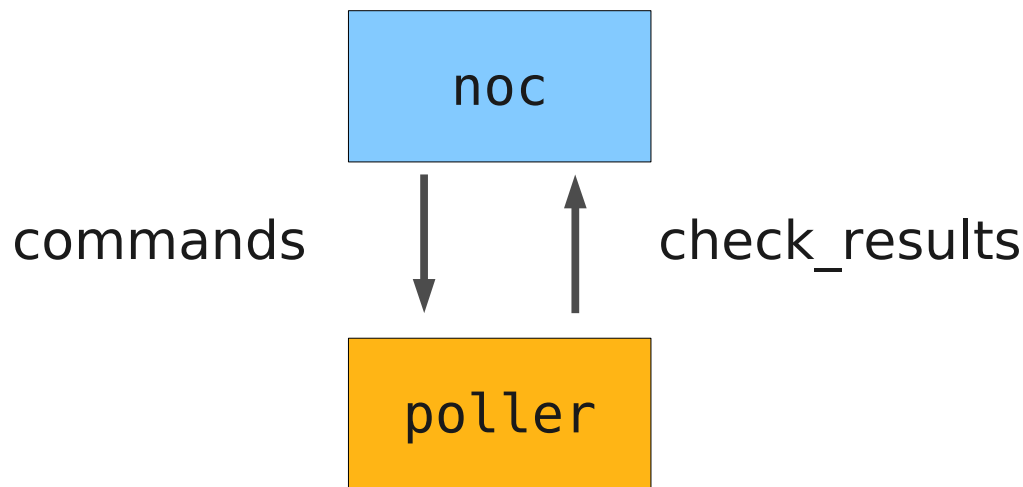
- Einer für Alle: NOC (Zentrale)
- Alle für Einen: Poller (Sender)
- Ein Nagios-Host kann gleichzeitig NOC und Poller sein!
- Multi-Tier



Merlin: Big Picture

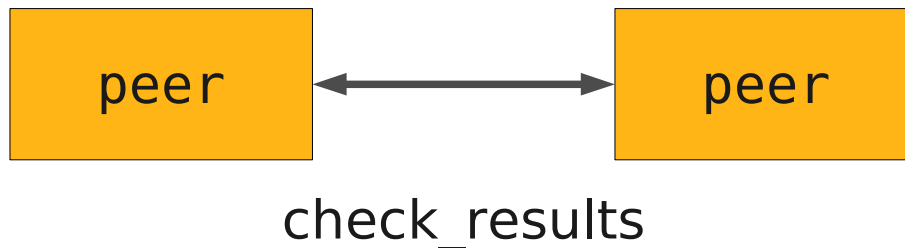


Houston, wir haben ein ...



Konfiguration:

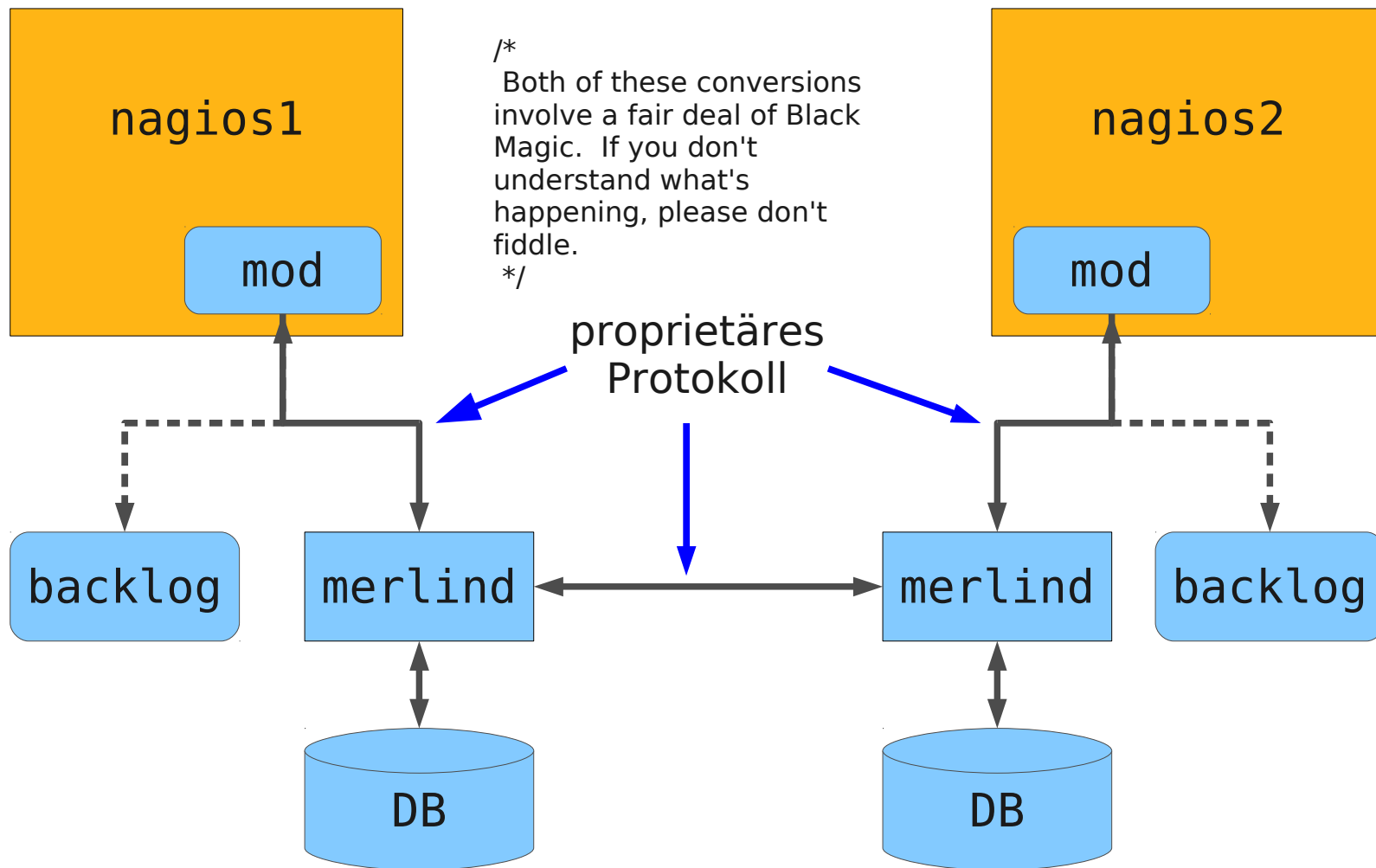
- 1:1 ?
- manuell



Konfiguration:

- 1:1
- manuell

Dämonisches ...



Nagios Event Broker

- Module abonnieren Channels
 - ◊ einklinken von Callback-Routinen
- Callbacks blockieren Nagios
 - ◊ so schnell wie möglich wieder zu Nagios zurückgeben
 - ◊ Daemon statt direktes Schreiben in die Datenbank
- Callbacks = ReadOnly
 - ◊ One-Way: Information aus Nagios
 - ◊ Schreiben: globale Nagios-Datenstrukturen modifizieren

Installation (1)

- Git verwenden:
 - `git clone git://git.op5.org/nagios/merlin.git`
 - Tar auf `www.op5.org` ist selten aktuell!
 - `git checkout master|next`
 - `git tag -l`
- Aktuell: v0.6.8 (2010-05-07)
- Webseite: v0.6.7 - hinkt ständig hinter

Installation (2)

- Kompilieren: benötigt libdbi!
 - ♦ Debian:

```
aptitude install libdbd-mysql \  
libdbi0-dev php5-cli php5-mysql
```
- make
 - ♦ ... aber nicht: `make install!`

Installation (3)

- Installskript benötigt passwortlosen Zugriff als root zu MySQL:
 - ~/.my.cnf verwenden
 - Test: „mysql“ als root

```
bash install-merlin.sh \  
  --nagios-cfg=/etc/nagios3/nagios.cfg \  
  --dest-dir=/usr/local/nagios/addons/merlin \  
  --db-type=mysql --db-user=merlin \  
  --db-pass=merlin --db-name=merlin
```

Konfiguration: merlin.conf

```
ipc_socket = /var/run/merlin/ipc.sock;
module {
    log_file = /var/log/merlin/neb.log;
    log_level = warn;
}
daemon {
    pidfile = /var/run/merlin/merlin.pid;
    log_level = warn;
    log_file = /var/log/merlin/daemon.log;
    import_program = php /usr/lib/merlin/import.php;
    port = 15551;
    database {
        name = merlin;
        user = merlin;
        pass = mostlysecret;
        host = localhost;
        type = mysql;
    }
}
```

Konfiguration: noc-poller

- chief (=noc):

```
poller slave {  
    address    = 192.168.1.1;  
    port       = 15551;  
    hostgroup  = SLAVE_SHOULD_WATCH_ON;  
}
```

- slave (=poller):

```
noc chief {  
    address    = 192.168.1.2;  
    port       = 15551;  
}
```

Konfiguration: peer

- peer:

```
peer other {  
    address    = 192.168.1.3;  
    port       = 15551;  
}
```

- jeder Host kann gleichzeitig peer, noc und poller sein
- maximal 65534 neighbours haben
neighbour = poller | noc | peer

merlin.conf - backlog

```
ipc_socket = /var/run/merlin/ipc.sock;
ipc_binlog_dir = /var/spool/merlin/backlogs;
module {
    ...
}
daemon {
    ...
}
```

- Die Backlog-Logik wird im Augenblick überarbeitet

Merlins Datenbank

- Warum überhaupt eine Datenbank?
 - Weil's einfach ist
 - Weil die NDOutils ein furchtbares Datenbanklayout haben
 - anständiges GUI
- Prinzip:
 - ein Objekt, eine Tabelle
 - keine historischen Daten (außer: state-changes, program start/stop, downtimes)
 - M:N-Relationen in eigener Tabelle, z.B. host_hostgroups

SELECT Service Status

- **NDOutils:**

```
SELECT nagios_instances.instance_id, nagios_instances.instance_name,  
nagios_services.host_object_id, obj1.name1 AS host_name,  
nagios_services.service_object_id, obj1.name2 AS service_description,  
nagios_servicestatus.* FROM `nagios_servicestatus`  
LEFT JOIN nagios_objects as obj1 ON  
nagios_servicestatus.service_object_id = obj1.object_id  
LEFT JOIN nagios_services ON nagios_servicestatus.service_object_id =  
nagios_services.service_object_id LEFT JOIN nagios_instances ON  
nagios_services.instance_id = nagios_instances.instance_id  
WHERE nagios_services.config_type = '1'  
ORDER BY instance_name ASC, host_name ASC, service_description ASC
```

- **Merlin:**

```
SELECT * FROM service ORDER by host_name, service_description ASC
```

MerlinDB (1): Tabellen

command, comment,
contact, contact_access, contact_contactgroup, contactgroup,
custom_vars, db_version, ~~downtime~~,
gui_access, gui_action_log,
host, host_contact, host_contactgroup, host_hostgroup
host_parents, hostdependency, hostescalation,
hostescalation_contact, hostescalation_contactgroup,
hostgroup,
notification, program_status,
report_data,
scheduled_downtime,
service, service_contact, service_contactgroup,
service_servicegroup, servicedependency, serviceescalation,
serviceescalation_contact, serviceescalation_contactgroup,
servicegroup,
timeperiod, timeperiod_exclude

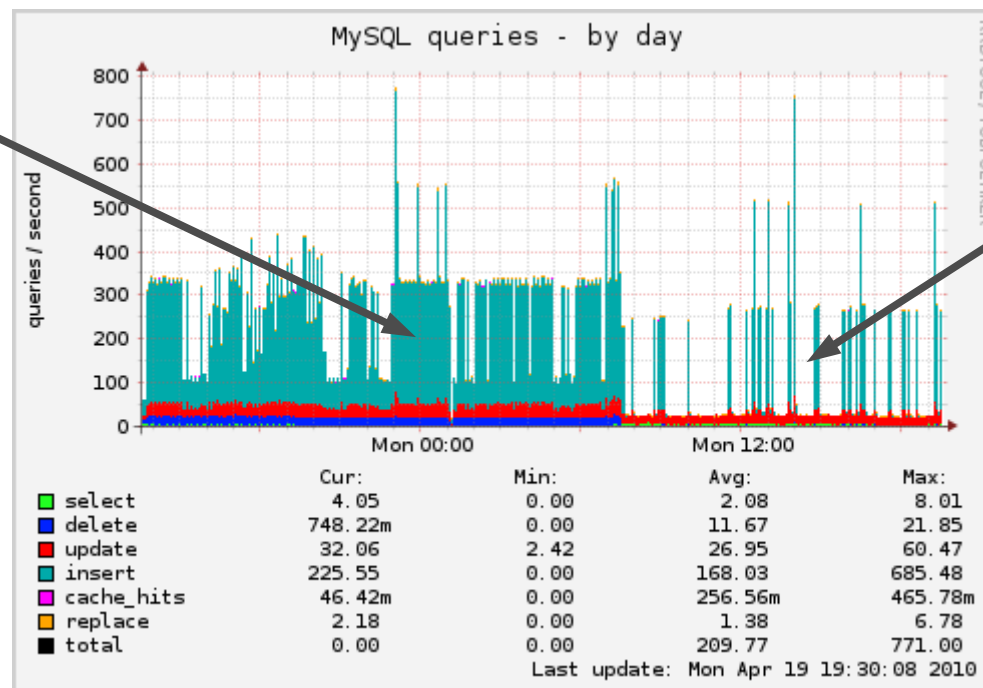
MerlinDB (2)

- Verknüpfung von Tabellen:
... where hosts.host_name =
 comments.host_name ...
- keine Objekt-IDs

Merlin als NDO-Ersatz

- Support durch NagVis
 - backendtype = „merlinmy“
- DB- Performance

ndo
+merlin



merlin

das wars ...

Vielen Dank
für Ihre Aufmerksamkeit.

Fragen?