Mod-Gearman

Distributed Monitoring based on the Gearman Framework

Sven Nierlein

24.05.2011

ConSol

Enterprising IT.

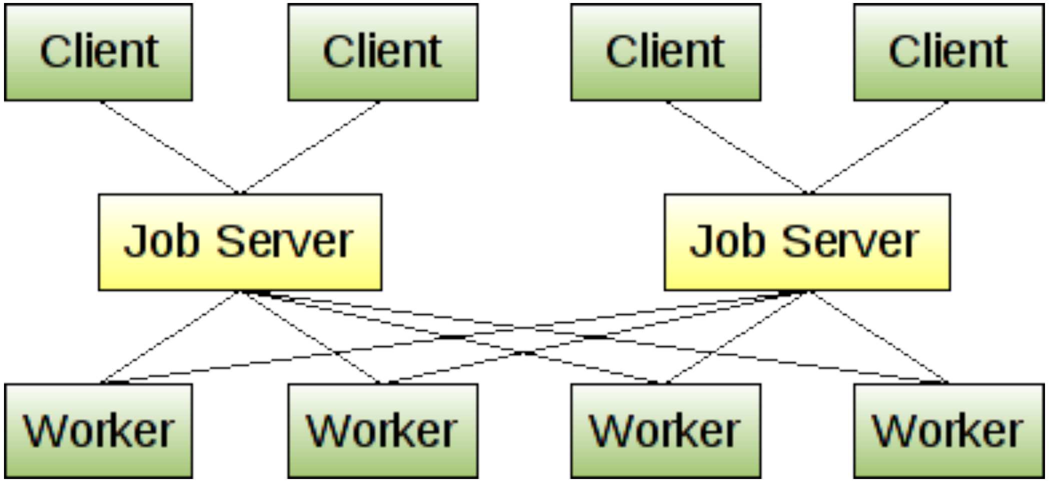
- Introduction
- Common Scenarios
- Configuration
- Performance Data
- Exports
- Tools
- OMD
- Hints

AGENDA

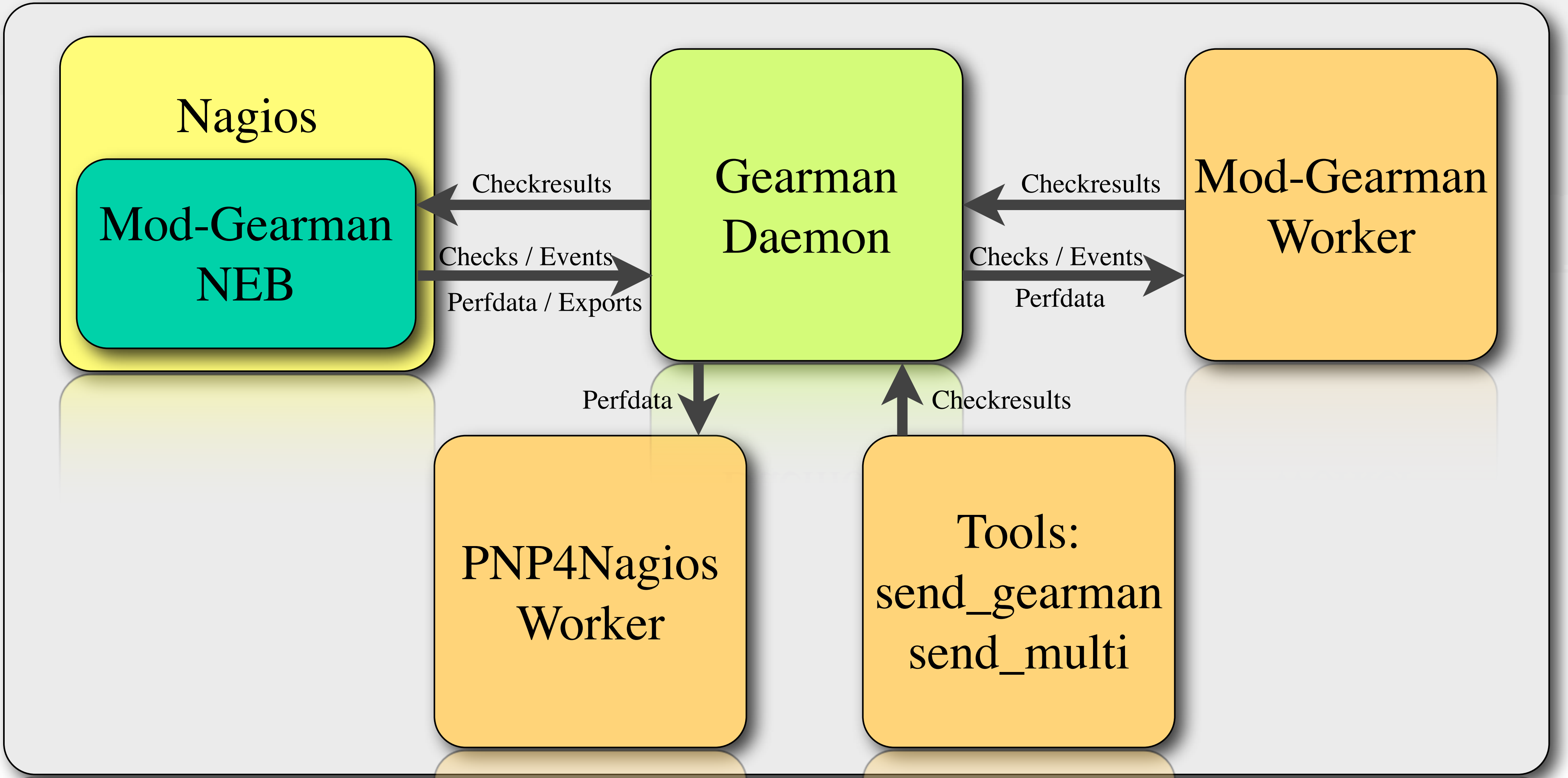
Introduction

Introduction

- **Gearman**
  - Distributes tasks across the network from multiple clients to multiple worker
  - Load balancing
  - Client/Worker supports C, Java, Perl, PHP, Python and Shell
  - Asynchronous

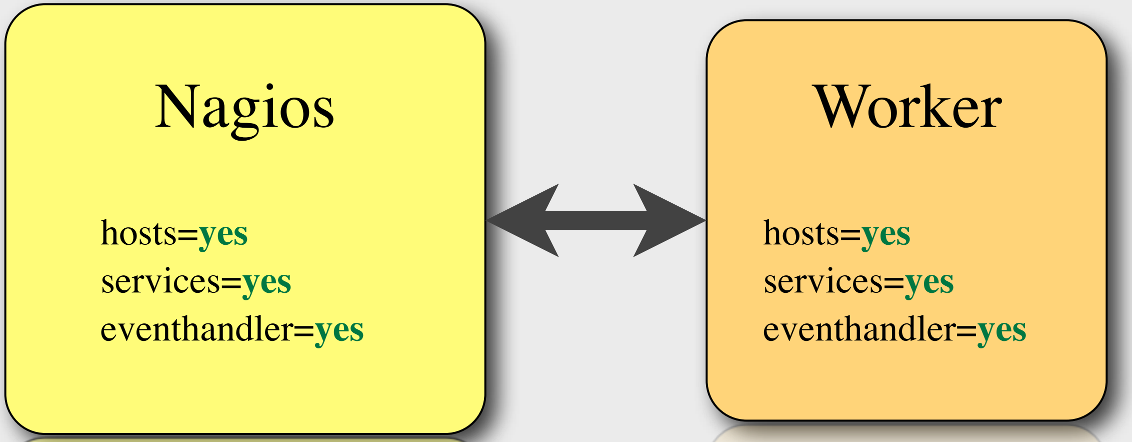


Introduction



Common Scenarios

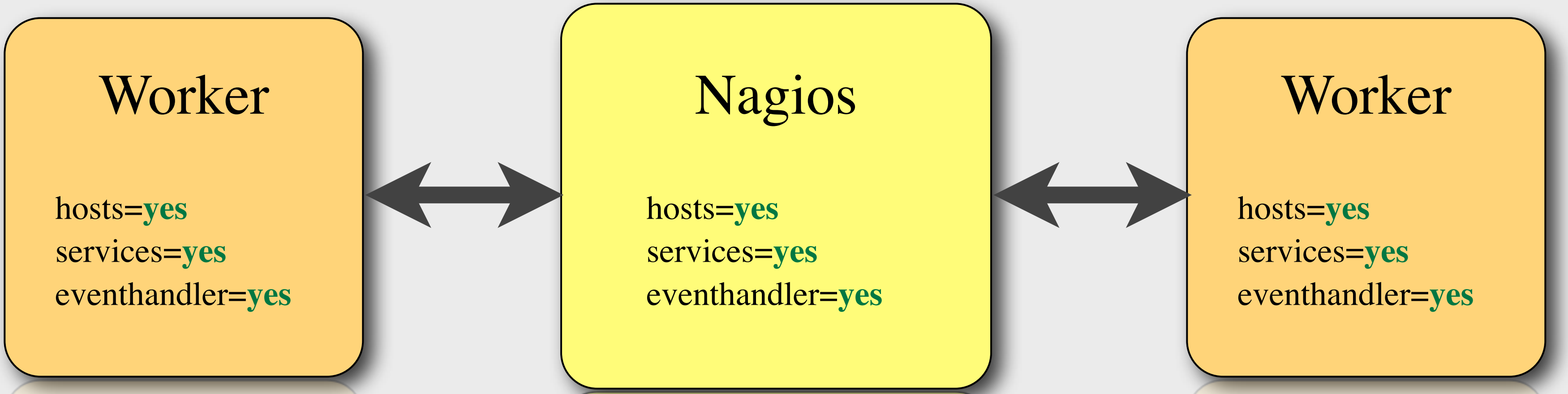
Load Reduction & Non Blocking



Pros

- Move blocking events away from Nagios core (Eventhandler, on-demand hostchecks)
- Reduce forking overhead from huge nagios core
- Even reduces load when both are on the same host

Load Balancing



Worker

hosts=yes
services=yes
eventhandler=yes

Nagios

hosts=yes
services=yes
eventhandler=yes

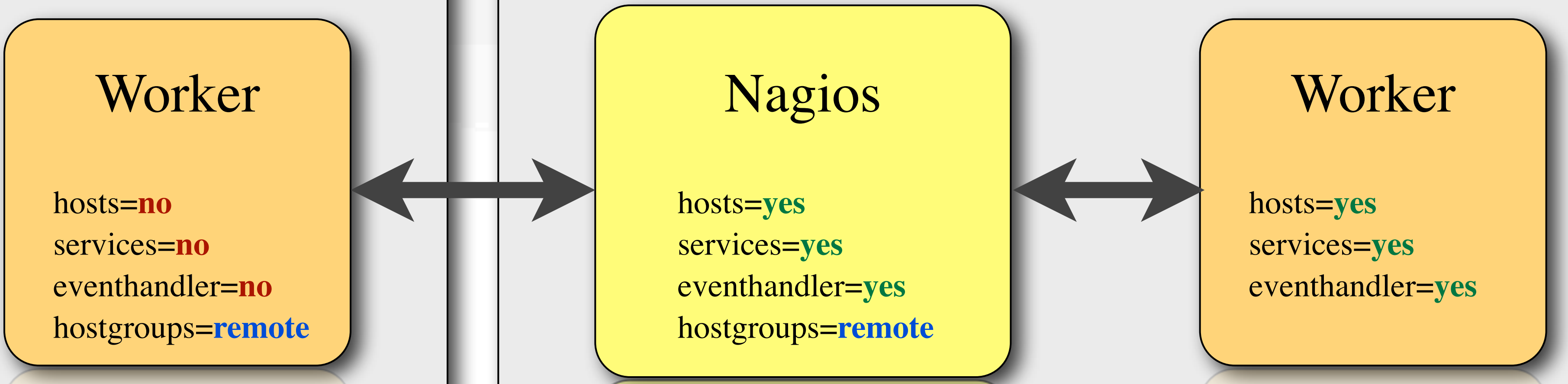
Worker

hosts=yes
services=yes
eventhandler=yes

Pros

- Spread load across multiple hosts

Distributed Setup



Worker

hosts=no
services=no
eventhandler=no
hostgroups=remote

Nagios

hosts=yes
services=yes
eventhandler=yes
hostgroups=remote

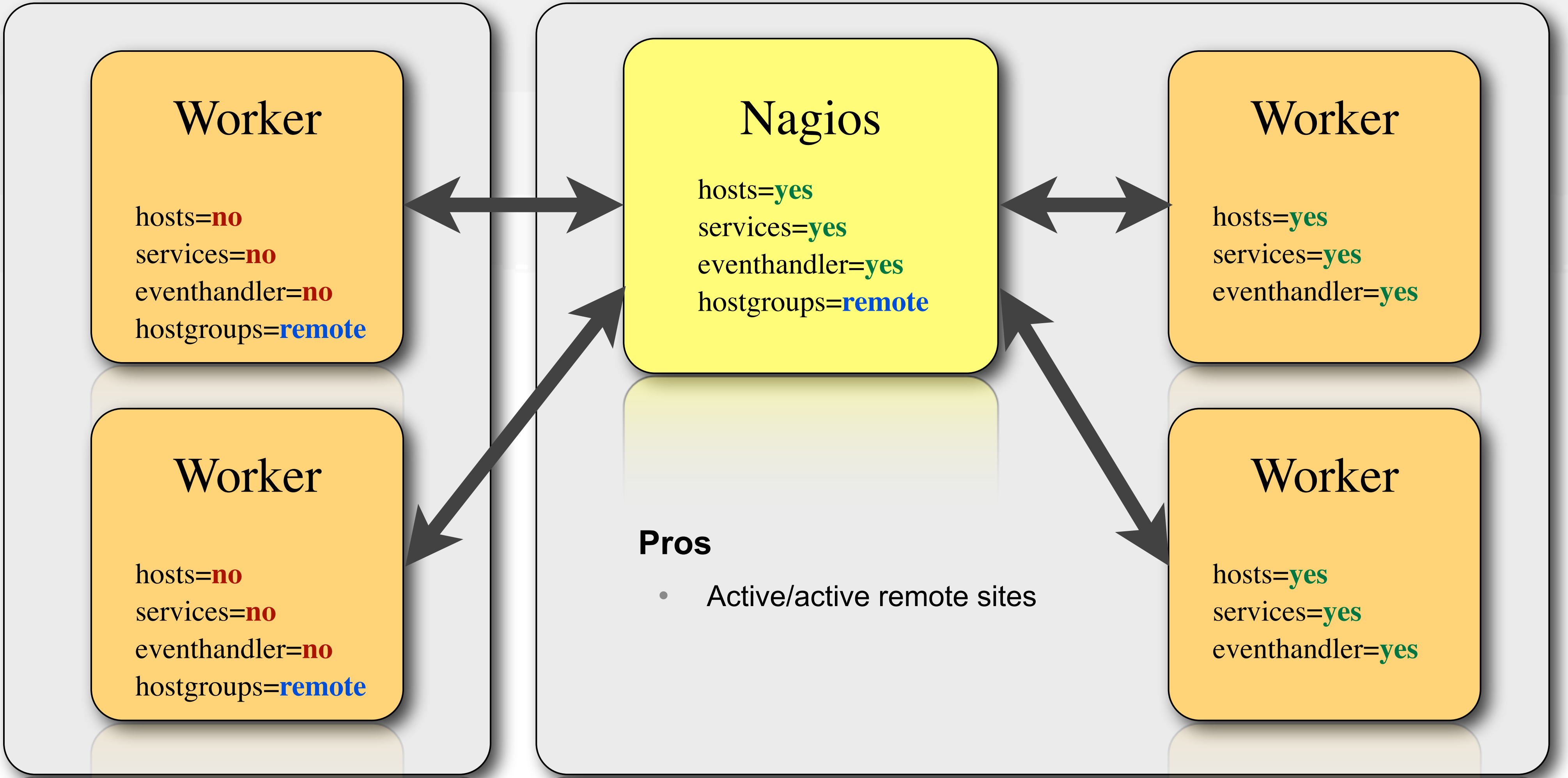
Worker

hosts=yes
services=yes
eventhandler=yes

Pros

- Easy replacement for remote nagios installations
- Central configuration

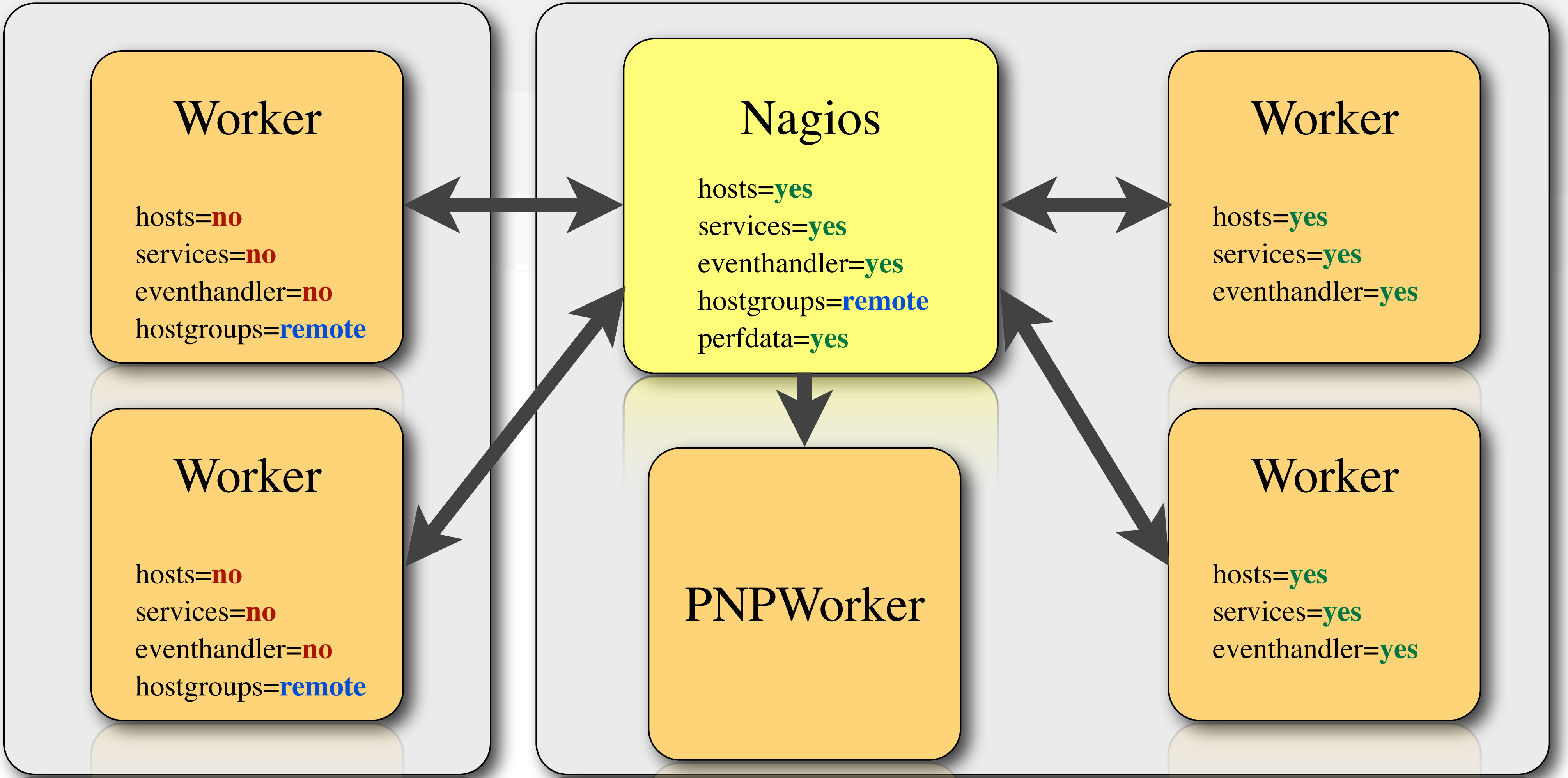
Distributed & Load Balancing



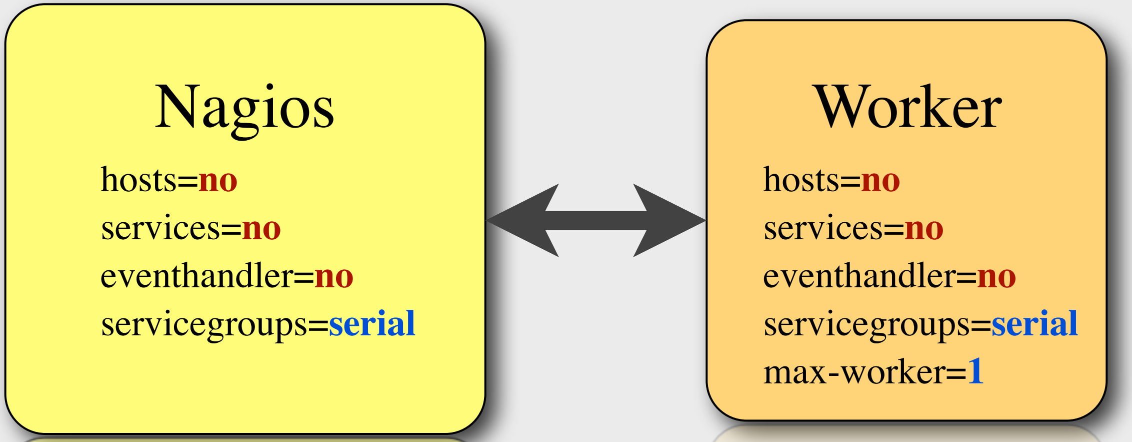
Pros

- Active/active remote sites

Distributed & Load Balancing + Graphing



Check Serialization



Nagios

hosts=**no**
services=**no**
eventhandler=**no**
servicegroups=**serial**

Worker

hosts=**no**
services=**no**
eventhandler=**no**
servicegroups=**serial**
max-worker=**1**

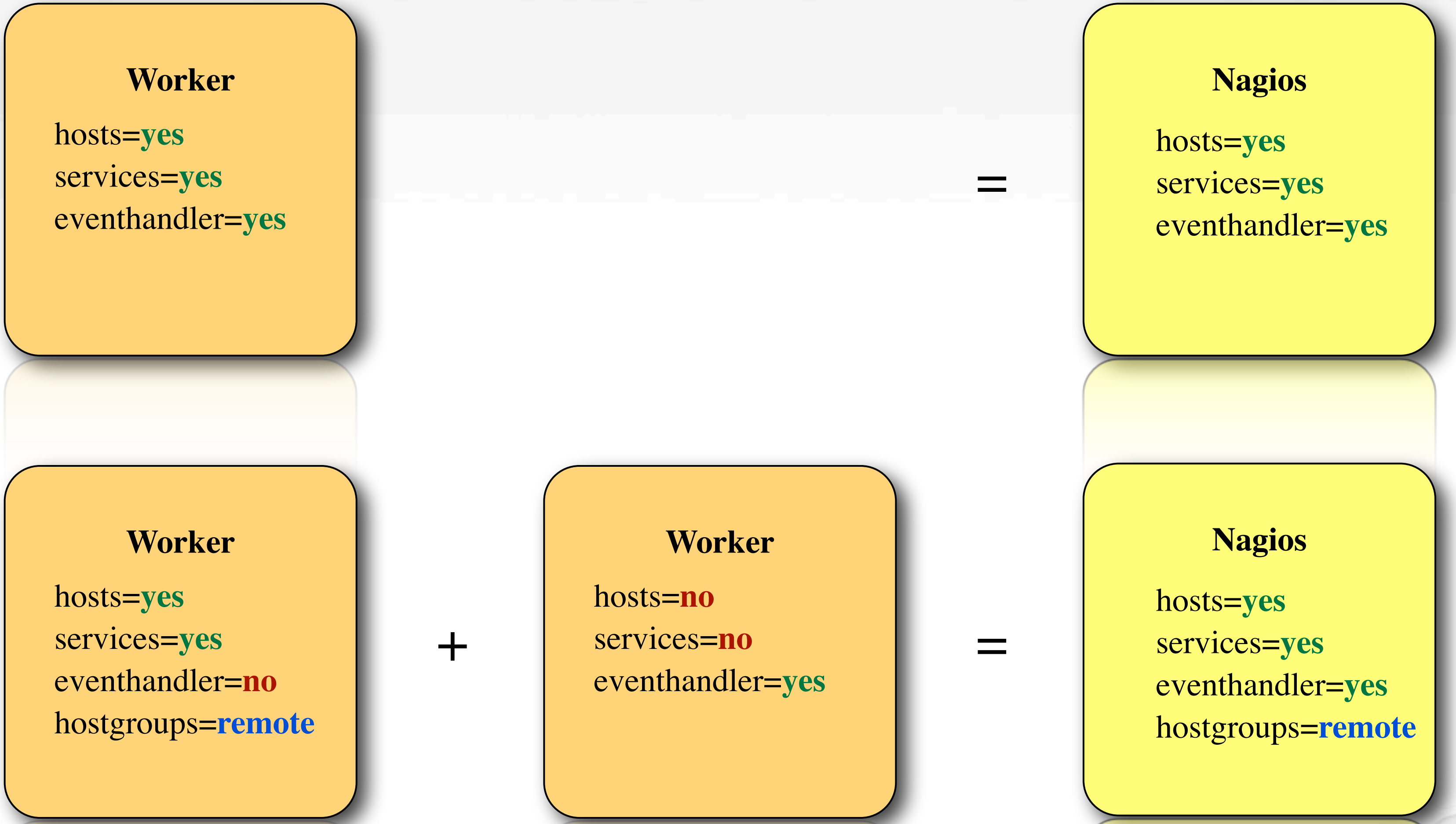
Pros

- Useful for non-serializable checks (ex. check_selenium, java checks. etc...)
- “parallelize_check” has been removed in Nagios 3.x
- Works better than “max_concurrent_checks”

Configuration

Configuration

- NEB configuration should be the sum of all workers



Worker

hosts=**yes**
services=**yes**
eventhandler=**yes**

=

Nagios

hosts=**yes**
services=**yes**
eventhandler=**yes**

Worker

hosts=**yes**
services=**yes**
eventhandler=**no**
hostgroups=**remote**

+

Worker

hosts=**no**
services=**no**
eventhandler=**yes**

=

Nagios

hosts=**yes**
services=**yes**
eventhandler=**yes**
hostgroups=**remote**

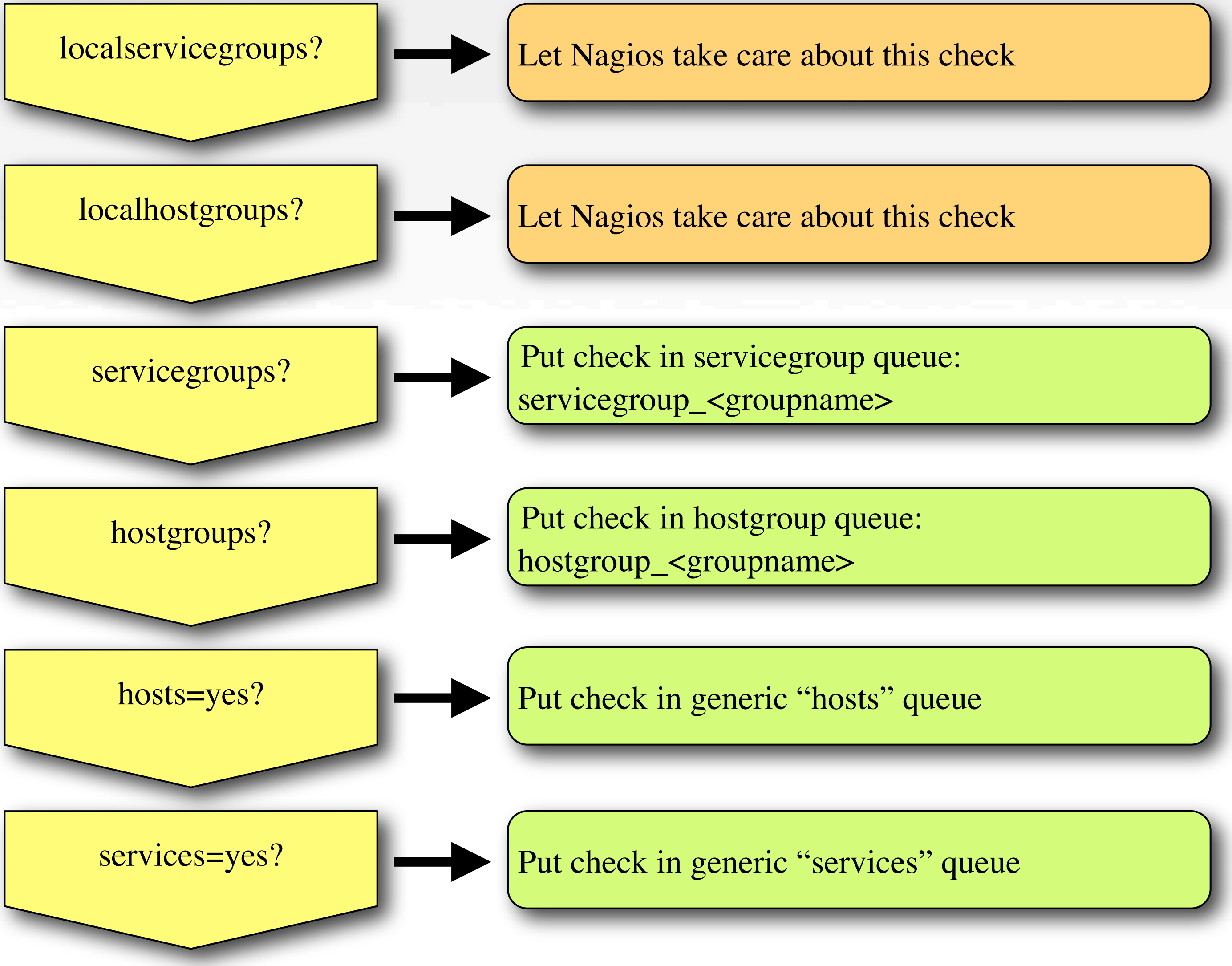
Configuration - Common

- **config**
  - can be used to specify/include config files
- **server**
  - list of gearmand servers to connect to
- **encryption**
  - enable/disable encryption
- **key**
  - plaintext key used for encryption
- **keyfile**
  - read key from this file

Configuration - Queues

- **services**
  - all servicechecks
- **hosts**
  - all hostchecks
- **hostgroups**
  - list of hostgroups going into a separate queue
- **servicegroups**
  - list of servicegroups going into a separate queue
- **eventhandler**
  - execute eventhandler with Mod-Gearman
- **localhostgroups**
  - list of hostgroups not managed by Mod-Gearman
- **localservicegroups**
  - list of servicegroups not managed by Mod-Gearman
- **do_hostchecks**
  - can be used to manage hostchecks by Nagios

Configuration - Queues



localservicegroups? → Let Nagios take care about this check

localhostgroups? → Let Nagios take care about this check

servicegroups? → Put check in servicegroup queue: servicegroup_<groupname>

hostgroups? → Put check in hostgroup queue: hostgroup_<groupname>

hosts=yes? → Put check in generic “hosts” queue

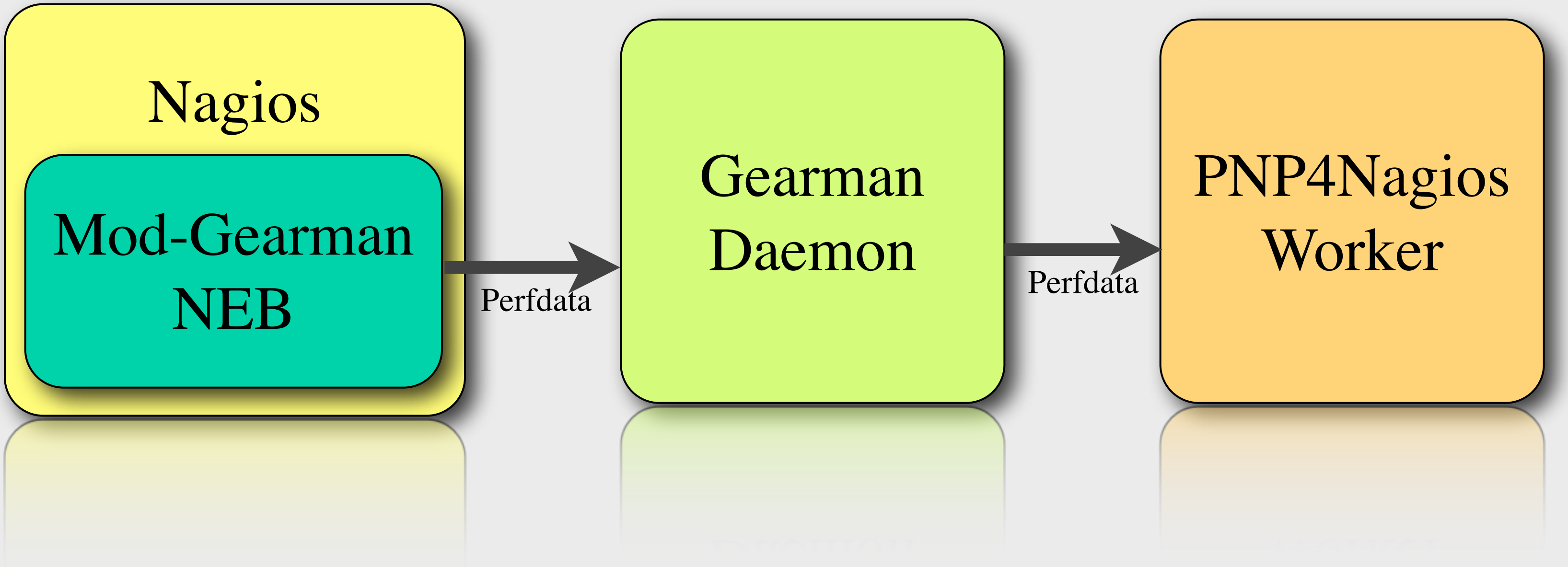
services=yes? → Put check in generic “services” queue

Configuration - Worker

- **identifier**
  - unique name of this worker, defaults to hostname
- **min-worker**
  - minimum number of total worker
- **max-worker**
  - maximum number of total worker
- **spawn-rate**
  - rate at which new worker will be spawned
- **idle-timeout**
  - timeout in seconds before a idling worker exists
- **max-jobs**
  - maximum number of jobs before a worker exists
- **dupserver**
  - useful to send copy of result to other Gearmand server

Performance Data

Performance Data



Config

- Set “perfdata=yes” in your Mod-Gearman neb configuration.
- Set “process_performance_data=1” in your nagios.cfg.
- Adjust gearman options in process_perfdata.cfg and start pnp_gearman_worker.

Exports

Exports

- **Export core events and data into gearman queues**
- **Format is JSON**
- **Write worker in any language gearman supports (C, Java, Perl, PHP, Python and Shell)**
- **No need to poll for data all the time**

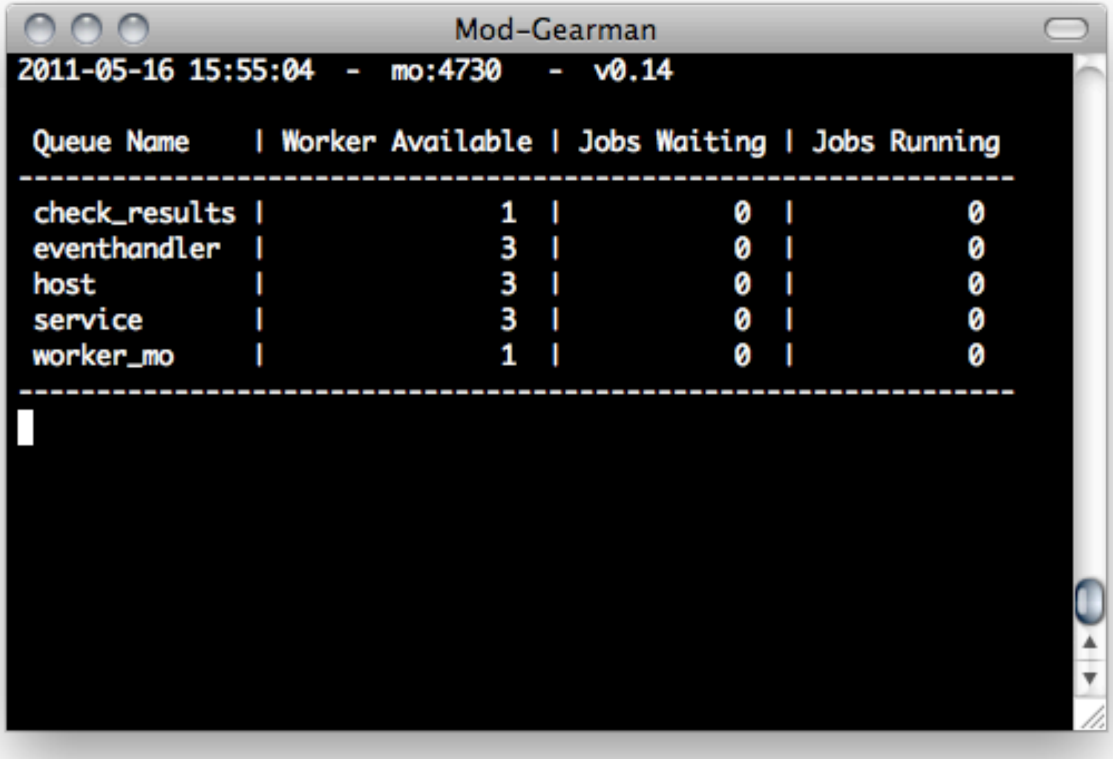
- **Example**
  - Syntax:
    export=<queue>:<returncode>:<callback>[,<callback>,...]
  - mod_gearman_neb.cfg:
    export=log_queue:1:NEBCALLBACK_LOG_DATA

- **Currently experimental and limited to a few callbacks:**
  - NEBCALLBACK_PROCESS_DATA
  - NEBCALLBACK_TIMED_EVENT_DATA
  - NEBCALLBACK_LOG_DATA

Tools

gearman_top

- **Shows current state of all queues**
  - $ gearman_top -H localhost:4730



```
2011-05-16 15:55:04  -  mo:4730   -  v0.14

 Queue Name     | Worker Available | Jobs Waiting | Jobs Running
-----------------------------------------------------------------
 check_results |                1  |            0  |            0
 eventhandler  |                3  |            0  |            0
 host          |                3  |            0  |            0
 service       |                3  |            0  |            0
 worker_mo     |                1  |            0  |            0
-----------------------------------------------------------------
```

check_gearman

- **Use as nagios plugin to check gearmand and worker**

  -
    ```
    $ ./check_gearman -H localhost
    check_gearman CRITICAL - failed to connect to localhost:4730 - Connection refused
    ```

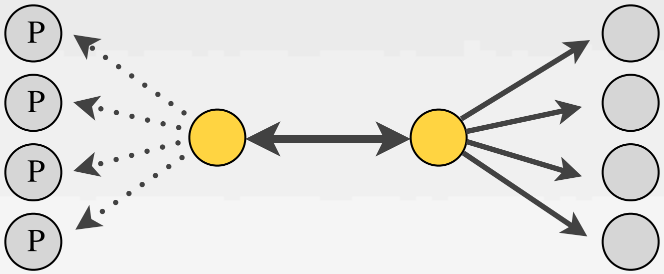
  -
    ```
    $ ./check_gearman -H localhost
    check_gearman OK - 0 jobs running and 0 jobs waiting. Version: 0.14|...
    ```

send_gearman

- **Similar but extended functionality like send_nsca**
- **Can be used to send passive check result via Mod-Gearman**
- **Can send active results with --active**
- **Use --latency, --starttime, --finishtime to preserve those attributes too**

  -
    ```
    $ ./bin/send_gearman --server=mo --keyfile=etc/mod-gearman/secret.key \
                                  --host='localhost' --service='ping' --message='Ping OK' --returncode=0
    ```

send_multi



- **Return multiple results from check_multi**
  - Basically:

    $ check_multi -r 256 -f check.cfg | ./bin/send_multi --config=mod_gearman.cfg --host=<host>

  - Better multi.sh:

```
#!/bin/bash
host=$1; shift;
other=$*
report="256"
if [ "$other" != "" ]; then
  report="13"
fi
out=`.../libexec/check_by_ssh -H $host -q -C ".../check_multi -f .../multi.cfg -r $report $other" 2>&1`
rc=$?
if [ `echo "$out" | grep -c "CHILD"` -eq 0 -o "$other" != "" ]; then
  echo "$out"
  exit $rc
fi
echo "$out" | .../send_multi config=.../mod_gearman.conf host=$host
```

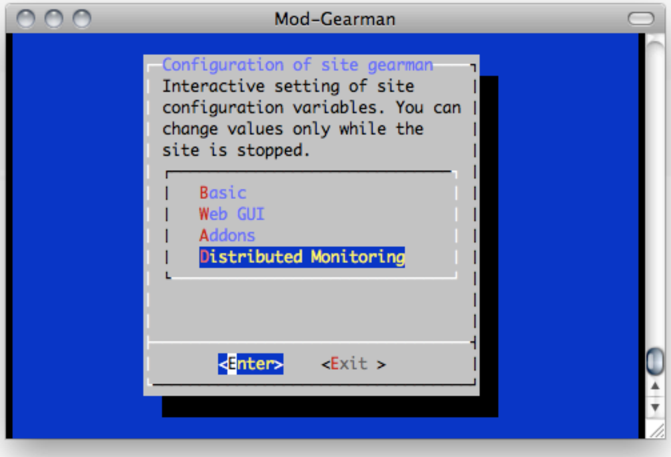
  - „check_multi -i <subcheck>” allows you to reschedule single checks from a multi.cfg

```
$ ./multi.sh                # for all
$ ./multi.sh -i check17     # for a single check
```

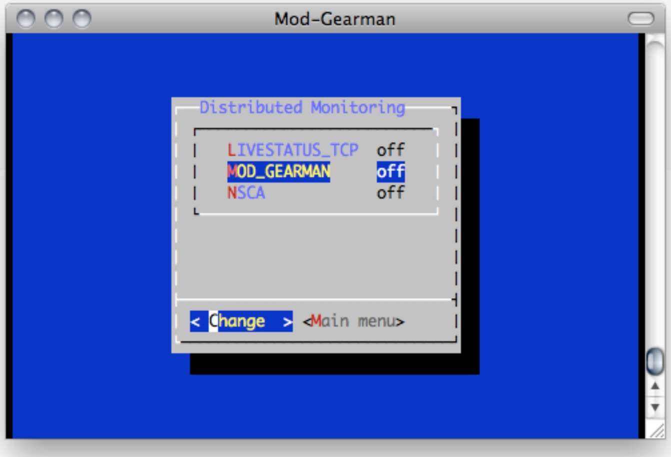
OMD

OMD

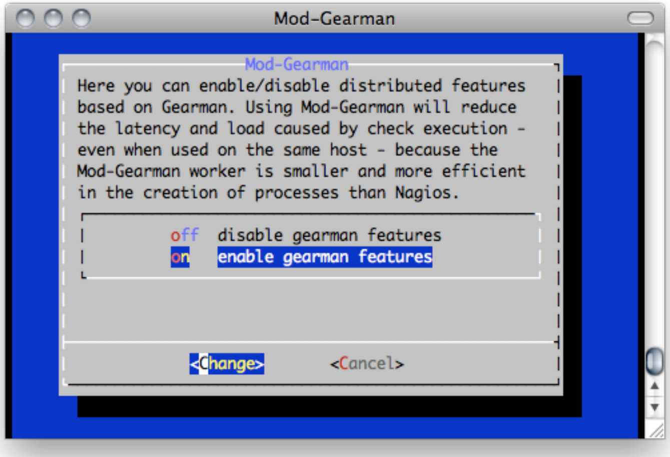
- **Mod-Gearman can be enabled via “omd config”**



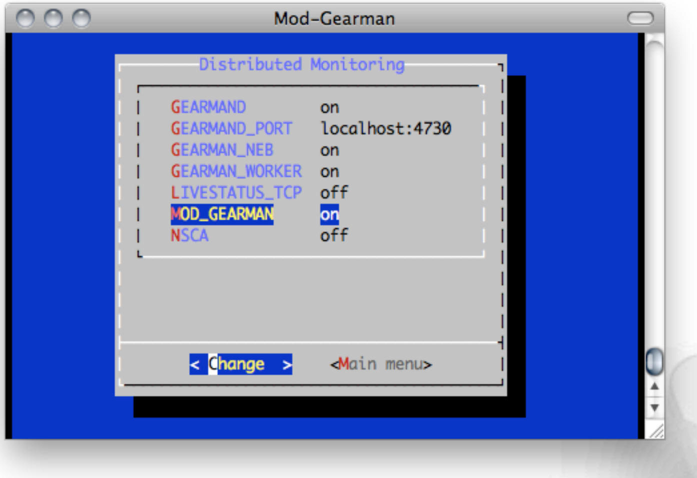
```
Configuration of site gearman
Interactive setting of site
configuration variables. You can
change values only while the
site is stopped.

    Basic
    Web GUI
    Addons
    Distributed Monitoring

        <Enter>     <Exit >
```



```
Distributed Monitoring

    LIVESTATUS_TCP   off
    MOD_GEARMAN      off
    NSCA             off

< Change  >  <Main menu>
```



```
Mod-Gearman
Here you can enable/disable distributed features
based on Gearman. Using Mod-Gearman will reduce
the latency and load caused by check execution -
even when used on the same host - because the
Mod-Gearman worker is smaller and more efficient
in the creation of processes than Nagios.

          off   disable gearman features
          on    enable gearman features

         <Change>      <Cancel>
```



```
Distributed Monitoring

    GEARMAND          on
    GEARMAND_PORT     localhost:4730
    GEARMAND_NEB      on
    GEARMAND_WORKER   on
    LIVESTATUS_TCP    off
    MOD_GEARMAN       on
    NSCA              off

        < Change  >     <Main menu>
```

OMD

- **Configuration:**

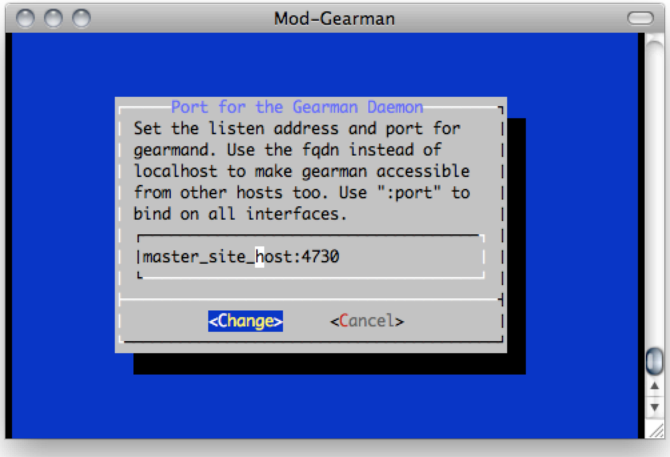
```
etc/mod-gearman/
├── nagios.cfg            # loading broker
├── perfdata.conf         # perfdata config part of server.cfg
├── port.conf             # tcp port for gearmand
├── secret.key            # encryption key
├── server.cfg            # neb module config
└── worker.cfg            # gearman worker config
```

- **Logfiles**

```
var/log/gearman/
├── gearmand.log
├── neb.log
└── worker.log
```

OMD

- **Connect multiple OMD instances**
- **Share the secret.key**
  - Use same secret.key for all connected OMD sites
  - /omd/sites/<site>/etc/mod-gearman/secret.key
  - Disable gearmand on remote workers
  - Enter master sites fqdn for nodes and master as GEARMAN_PORT



Port for the Gearman Daemon

Set the listen address and port for gearmand. Use the fqdn instead of localhost to make gearman accessible from other hosts too. Use ":port" to bind on all interfaces.

```
master_site_host:4730
```

Hints

Hints

- **Always monitor your gearman infrastructure! (check_gearman)**
  - Put gearman infrastructure monitors into the “localservicegroups”.
- **Enable freshness checks**
- **Secure gearmand (ex.: iptables)**
  - gearmand currently has no access control

Resources

- http://labs.consol.de/nagios/mod-gearman/
- http://gearman.org/
- http://docs.pnp4nagios.org/de/pnp-0.6/modes#gearman_mode
- http://my-plugin.de/wiki/projects/check_multi/feed_passive
- http://packages.debian.org/de/source/sid/mod-gearman

Questions?