Storage Management using Nagios®

Panagiotis Garefalakis
ICS-FORTH
Heraklion, Greece

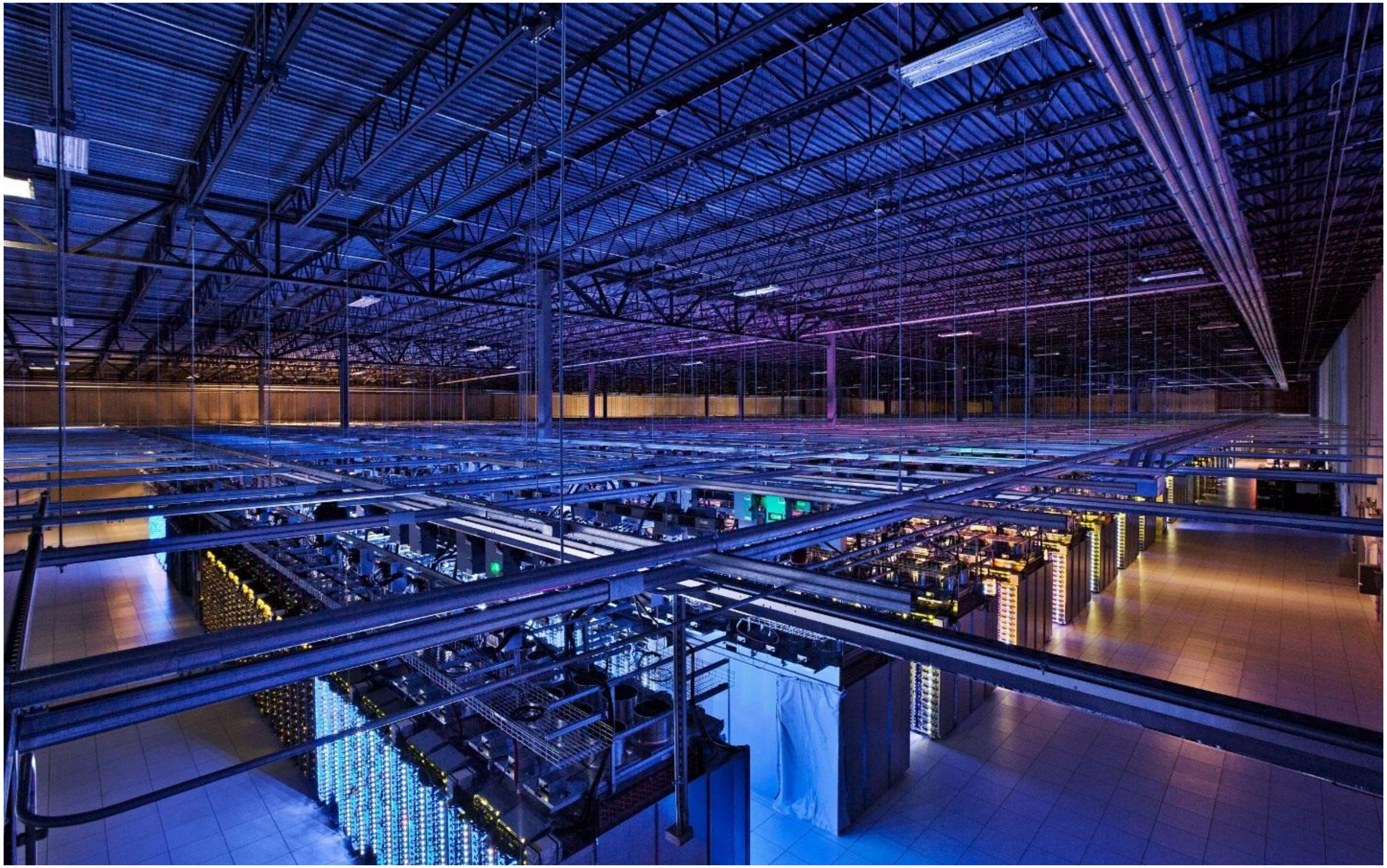
FORTH
Institute of Computer Science

pgaref@ics.forth.gr

Outline

- Storage management challenges
- What is Nagios
- Tutorial topics:
  - How to start a Nagios server
  - Writing storage service monitoring code
  - Monitoring local & remote storage
  - Event handling

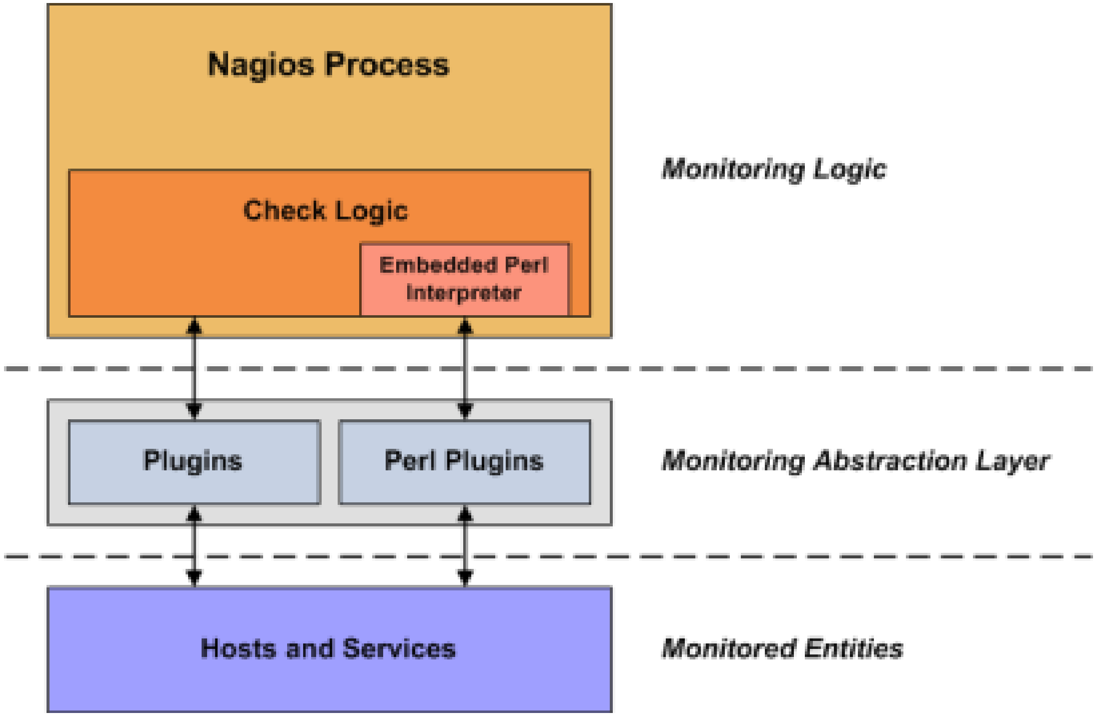
Storage management challenges



What is Nagios?

- A key measurement tool for actively monitoring availability of devices and services.
- The most used open source network monitoring software.
- Can support monitoring and management of thousands of devices and services.

Nagios: an open-source system for distributed monitoring, and control



Generic Host

```
define host{
     name                       generic-host
     notifications_enabled          1
     event_handler_enabled          1
     flap_detection_enabled         1
     process_perf_data              1
     retain_status_information      1
     retain_nonstatus_information   1
     check_command                  check-host-alive
     max_check_attempts             5
     notification_interval          60
     notification_period            24x7
     notification_options           d,r
     contact_groups                 nobody
     register                       0
     }
```

Individual Host

```
define host{
        use                     generic-host
        host_name               switch1
        alias                   Core_switches
        address                 192.168.1.2
        parents                 router1
        contact_groups          switch_group
}
```

Generic Service

```
define service{
        name                            generic-service
        active_checks_enabled           1
        passive_checks_enabled          1
        parallelize_check               1
        obsess_over_service             1
        check_freshness                 0
        notifications_enabled           1
        event_handler_enabled           1
        flap_detection_enabled          1
        process_perf_data               1
        retain_status_information       1
        retain_nonstatus_information    1
        is_volatile                     0
        check_period                    24x7
        max_check_attempts          5
        normal_check_interval       5
        retry_check_interval            1
        notification_interval           60
        notification_period             24x7
        notification_options            c,r
        register                        0
        }
```

Individual Service

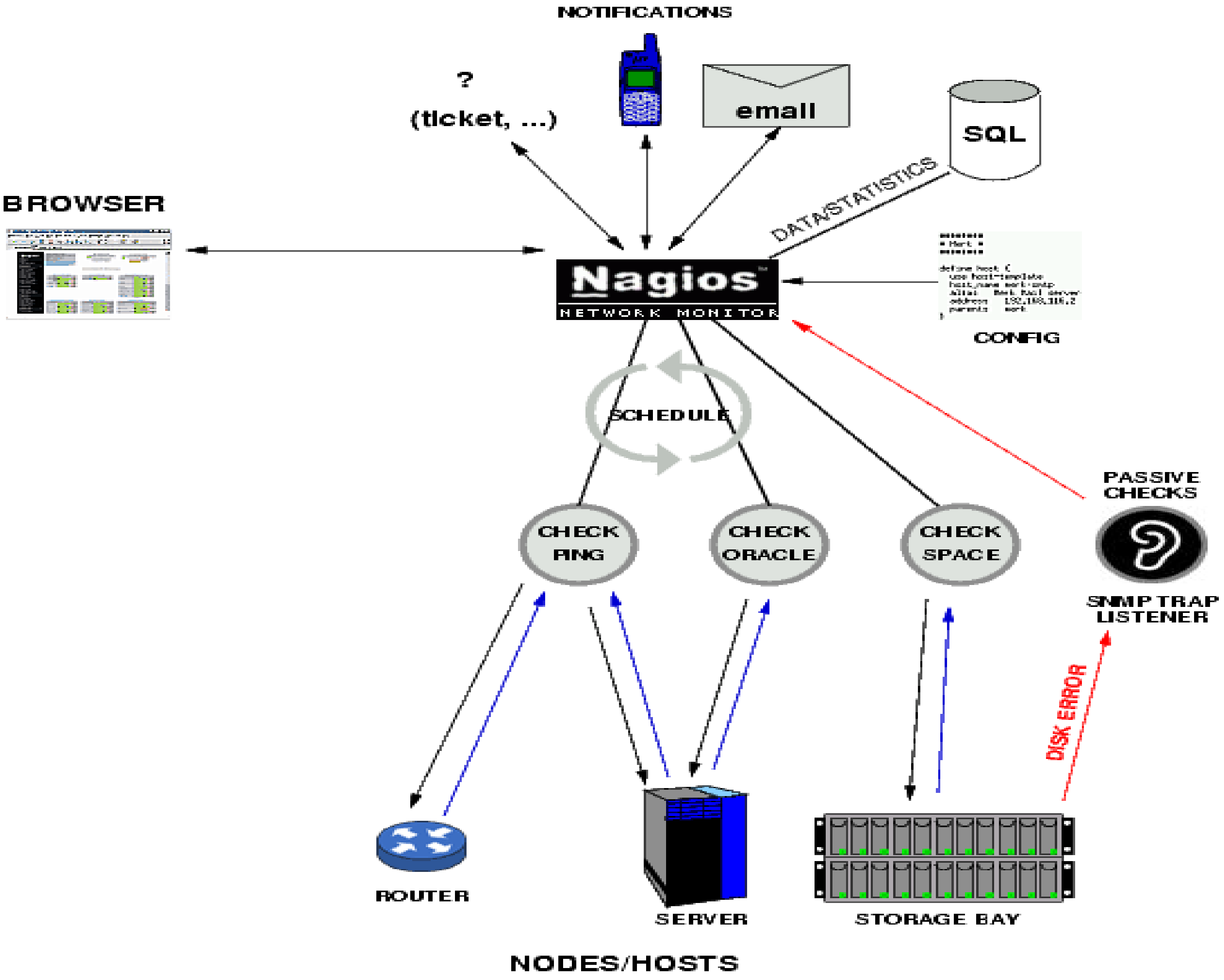
```
define service{
        host_name                       switch1
        use                             generic-service
        service_description             PING
        check_command                   check-host-alive
        max_check_attempts              5
        normal_check_interval           5
        notification_options            c,r,f
        contact_groups                  switch-group
}
```

Commands

- Commands wrap the check scripts.

```
define command{
        command_name    check-host-alive
        command_line    $USER1$/check_ping -H
 $HOSTADDRESS$ -w 99,99% -c 100,100% -p 1
        }
```

- Check scripts can be in any language.



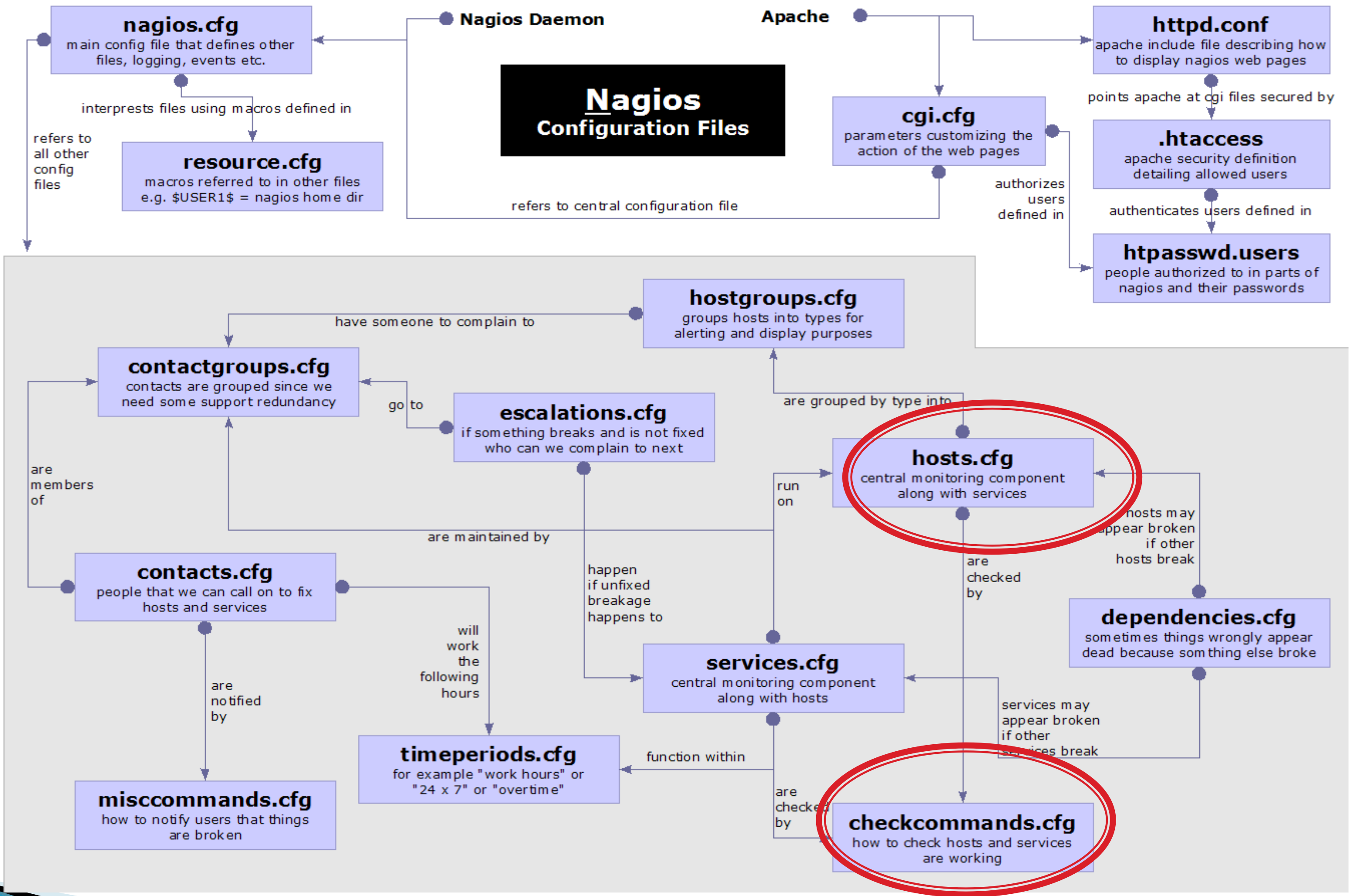
```
########
# [illegible] #
########

define host {
  use host-template
  host_name [illegible]
  alias     [illegible]
  address   192.168.116.2
  parents   [illegible]
}
```

Installing Nagios Core (the server)

- Manual install.

- Read the installation instructions (USB).
  - Installation commands – Ubuntu
  - Installation script – CentOS

Configuration



Nagios Configuration Files

Nagios Daemon

Apache

nagios.cfg
main config file that defines other files, logging, events etc.

interprests files using macros defined in

refers to all other config files

resource.cfg
macros referred to in other files e.g. $USER1$ = nagios home dir

refers to central configuration file

httpd.conf
apache include file describing how to display nagios web pages

points apache at cgi files secured by

cgi.cfg
parameters customizing the action of the web pages

authorizes users defined in

.htaccess
apache security definition detailing allowed users

authenticates users defined in

htpasswd.users
people authorized to in parts of nagios and their passwords

hostgroups.cfg
groups hosts into types for alerting and display purposes

have someone to complain to

contactgroups.cfg
contacts are grouped since we need some support redundancy

go to

escalations.cfg
if something breaks and is not fixed who can we complain to next

are grouped by type into

hosts.cfg
central monitoring component along with services

run on

are maintained by

are members of

hosts may appear broken if other hosts break

are checked by

contacts.cfg
people that we can call on to fix hosts and services

happen if unfixed breakage happens to

will work the following hours

dependencies.cfg
sometimes things wrongly appear dead because somthing else broke

services.cfg
central monitoring component along with hosts

are notified by

services may appear broken if other services break

function within

timeperiods.cfg
for example "work hours" or "24 x 7" or "overtime"

are checked by

misccommands.cfg
how to notify users that things are broken

checkcommands.cfg
how to check hosts and services are working

Running a plugin locally

- Locate Nagios configuration files. /usr/local/nagios/etc/objects
- Open localhost.cfg (Sudo access)
- Add the lines:

```
define service{
        use                             local-service        ; Name of service template to use
        host_name                       localhost
        service_description             DISK_TEST
        check_command                   check_local_disk!70%!20%!/dev/sda3
        }
```

- Validate: `sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg`
- Restart Nagios: /etc/init.d/nagios restart

Monitoring growth of files

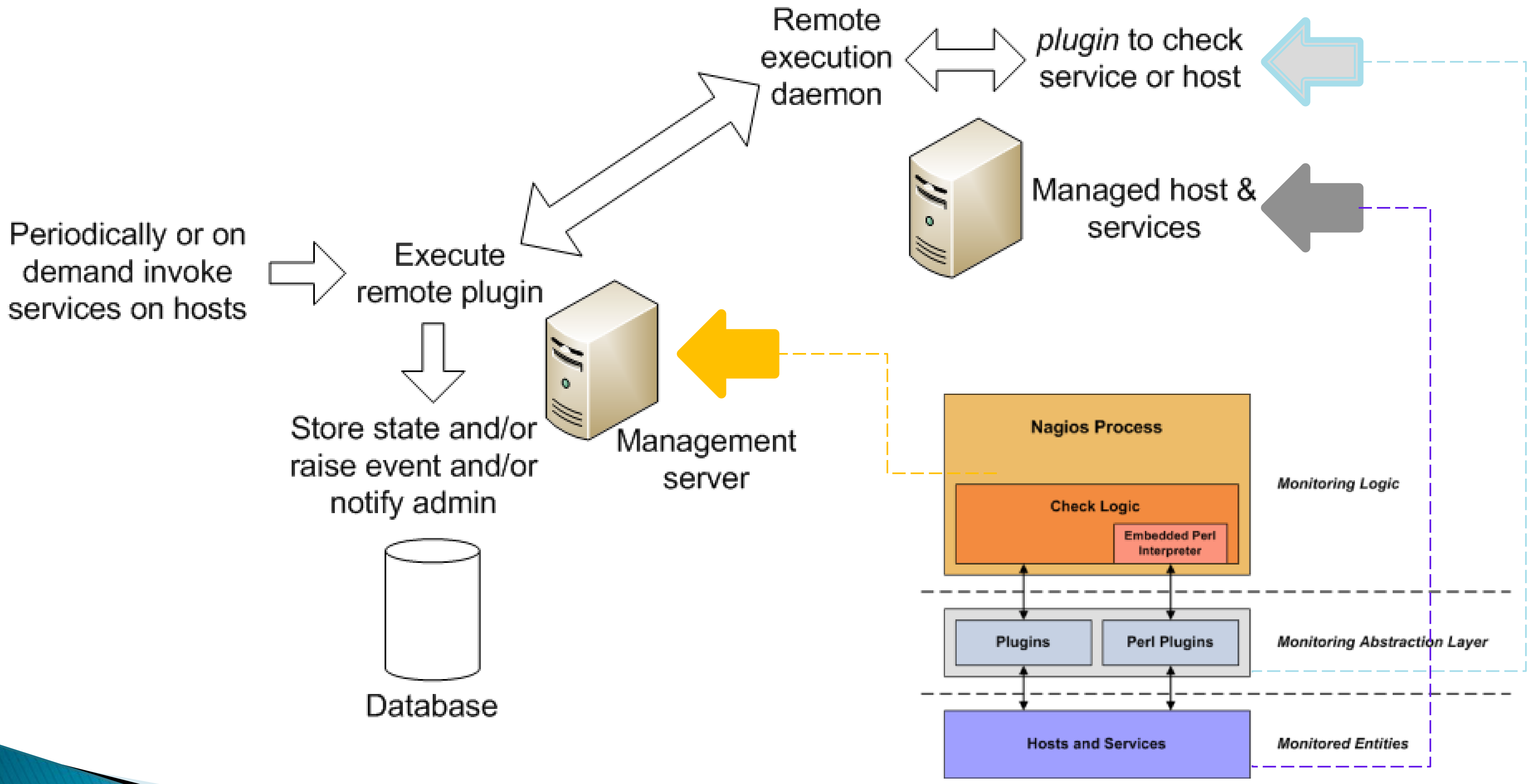
- Copy the given plugin “check_dir.sh” to nagios plugin directory:/usr/local/nagios/libexec
- Modify the commands.cfg file:

```
define command{
        command_name                    check_dir
        command_line                    $USER1$/check_dir $ARG1$
        }
```

- Modify the localhost.cfg file:

```
define service{
        use                             local-service
        host_name                       localhost
        service_description             CHECK_LOG
        check_command                   check_dir!9999
        }
```

Remote plugin execution: checking status of NFS server



Remote execution daemon

plugin to check service or host

Managed host & services

Periodically or on demand invoke services on hosts

Execute remote plugin

Store state and/or raise event and/or notify admin

Management server

Database

Nagios Process

Check Logic

Embedded Perl Interpreter

Plugins

Perl Plugins

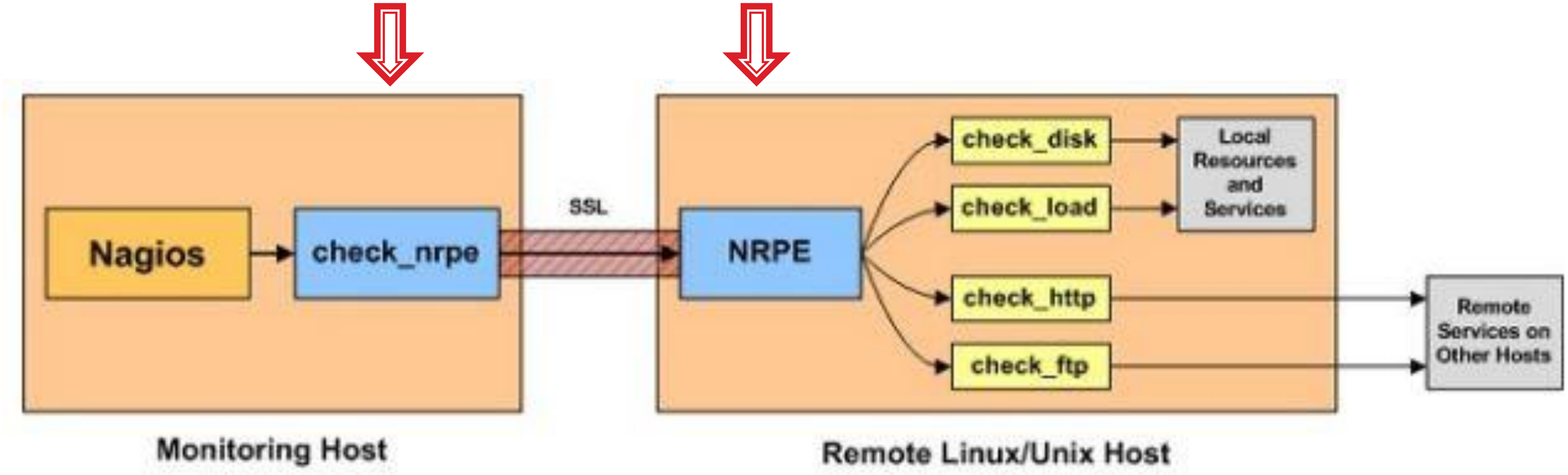
Hosts and Services

Monitoring Logic

Monitoring Abstraction Layer

Monitored Entities

NRPE

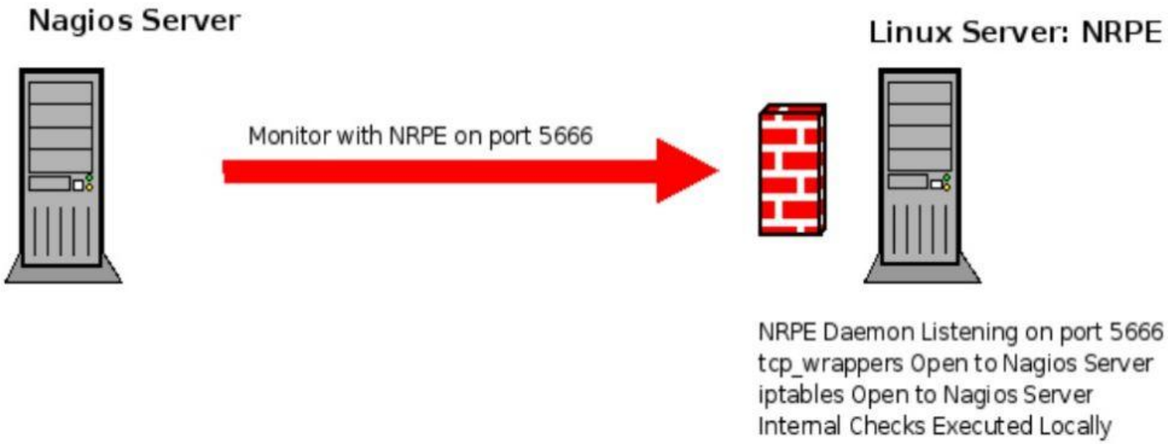


- ○Remote Host IP is: 139.91.70.76
- ○Your IP has to be added at nrpe.cfg before running!

Starting NRPE server

- Follow the instructions to install NRPE Server "Enable NRPE Server–Ubuntu.txt"
- You can Skip the Command and Service Definitions.
- You can check your connection by running the following command and using the IP Address of the remote box you want to monitor. You should get the return "NRPE v2.8.1" if all is working.
  - Command: `/usr/lib/nagios/plugins/check_nrpe –H 139.91.70.76`

Remote monitoring of NFS service


Nagios Server
Linux Server: NRPE
Monitor with NRPE on port 5666
NRPE Daemon Listening on port 5666
tcp_wrappers Open to Nagios Server
iptables Open to Nagios Server
Internal Checks Executed Locally

- We have an NFS server running in the remote host. A plugin for monitoring NFS is included “check_nfsmount.pl”
- We will modify NRPE configuration at the server part to be able to run check nfs remotely.
- Finally test the command:
  - /usr/lib/nagios/plugins/check_nrpe -H 139.91.70.76 -c check_nfs

Going further: Event handlers

- Nagios can attempt to rectify a fault by running a script.
- We can use Event Handlers to take action when something goes wrong.
  - Growing File example:
    - Print error message
    - Compress File
    - Truncate File

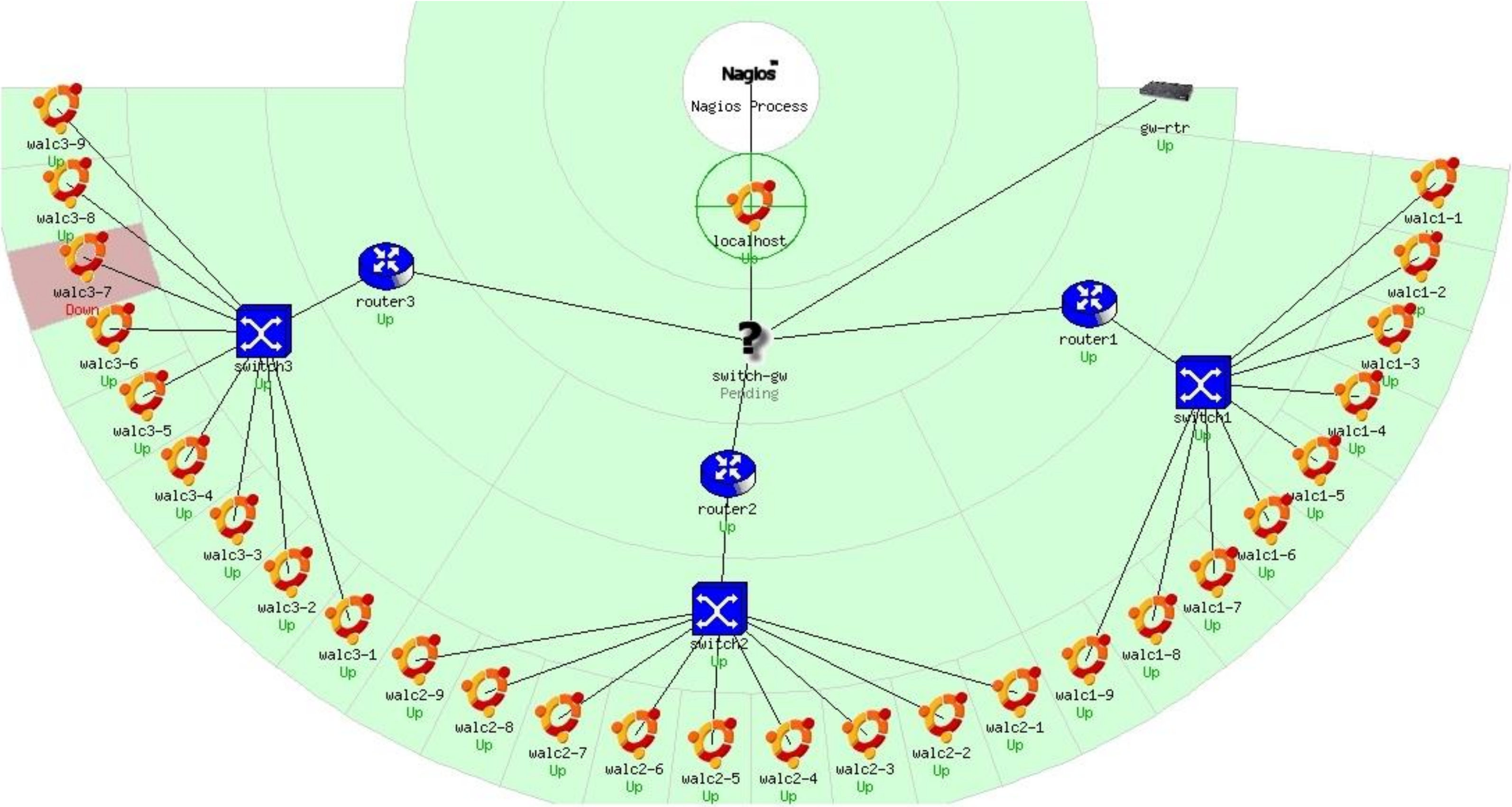
An event handler

- We want to react to above-threshold growth of files.
- Copy myhandler.sh to libexec/eventhandler
  - Change permission to nagios user!
- Add the following line to our command:
- event_handler my_handler!$SERVICESTATE$ $STATETYPE$ $SERVICEATTEMPT$

Finally add the command:

```
define command{
        command_name my_eventhandler
        command_line $USER1$/eventhandlers/myhandler  $ARG1$
}
```

Nagios web interface



Conclusions

- Nagios is a very useful tool saving time of administrators but can appear very complex when you first look at it.
- My advice is:
  - Install it on your test node (though this may well end up as your master server)
  - Run a few check scripts by hand to get the feel for them
  - Set up a simple config file that runs a few check on the local host
  - Install nrpe on the host and nrpe and nagios-plugins on a remote host
  - Run check nrpe by hand to get it working then add a couple of simple checks on the remote host
  - Now add hosts and service until you run out, then write some more

References

- http://www.nagios.org Nagios web site
- http://sourceforge.net/projects/nagiosplug Nagios plugins site
- http://www.nagiosexchange.org Unofficial Nagios plugin site
- http://www.debianhelp.co.uk/nagios.htm A Debian tutorial on Nagios
- http://www.nagios.com/ Commercial Nagios support