Mod-Gearman

Mike Weber

mweber@spidertools.com

Mod-Gearman

▶ **Basic Functionality**

What does Mod-Gearman do?
How does it work?

▶ **Self-Load Balancing**

Using Mod-Gearman to reduce load on XI.

▶ **Load Balancing**

Use workers to reduce load on Nagios.

▶ **Distributed Monitoring**

Specify load on numerous networks and workers.

▶ **Passive Monitoring**

Receiving passive checks.

Nagios
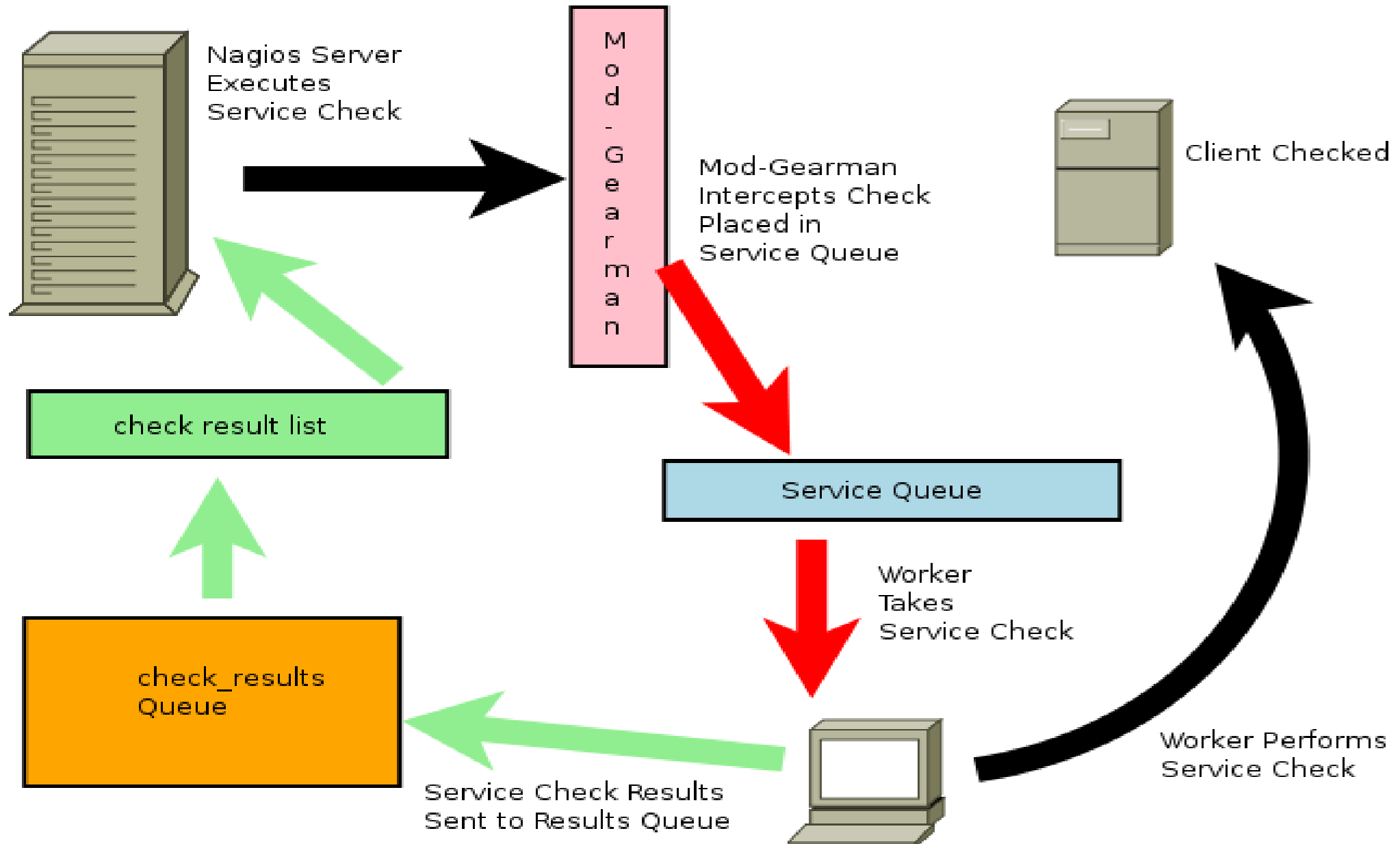**World Conference**
North America

Mod-Gearman: Basic Functionality

▶ **Distributes tasks to multiple workers**
Mutiple Servers
Multiple Workers

▶ **Supports Multiple Programming Languages:**
C
 Java
 Perl
 PHP
 Python
 Shell

Mod-Gearman

Nagios Server
Executes
Service Check

Mod-Gearman

Mod-Gearman
Intercepts Check
Placed in
Service Queue

Client Checked

check result list

Service Queue

Worker
Takes
Service Check

Worker Performs
Service Check

check_results
Queue

Service Check Results
Sent to Results Queue
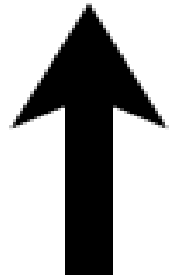
NEB: Nagios Event Broker

Nagios

User Creates an Application that accesses
Nagios Internals using NEB Modules

Nagios Events Captured

NEB Modules:
service checks
host checks
event handlers
comments
scheduled downtime
flapping
contact notification change
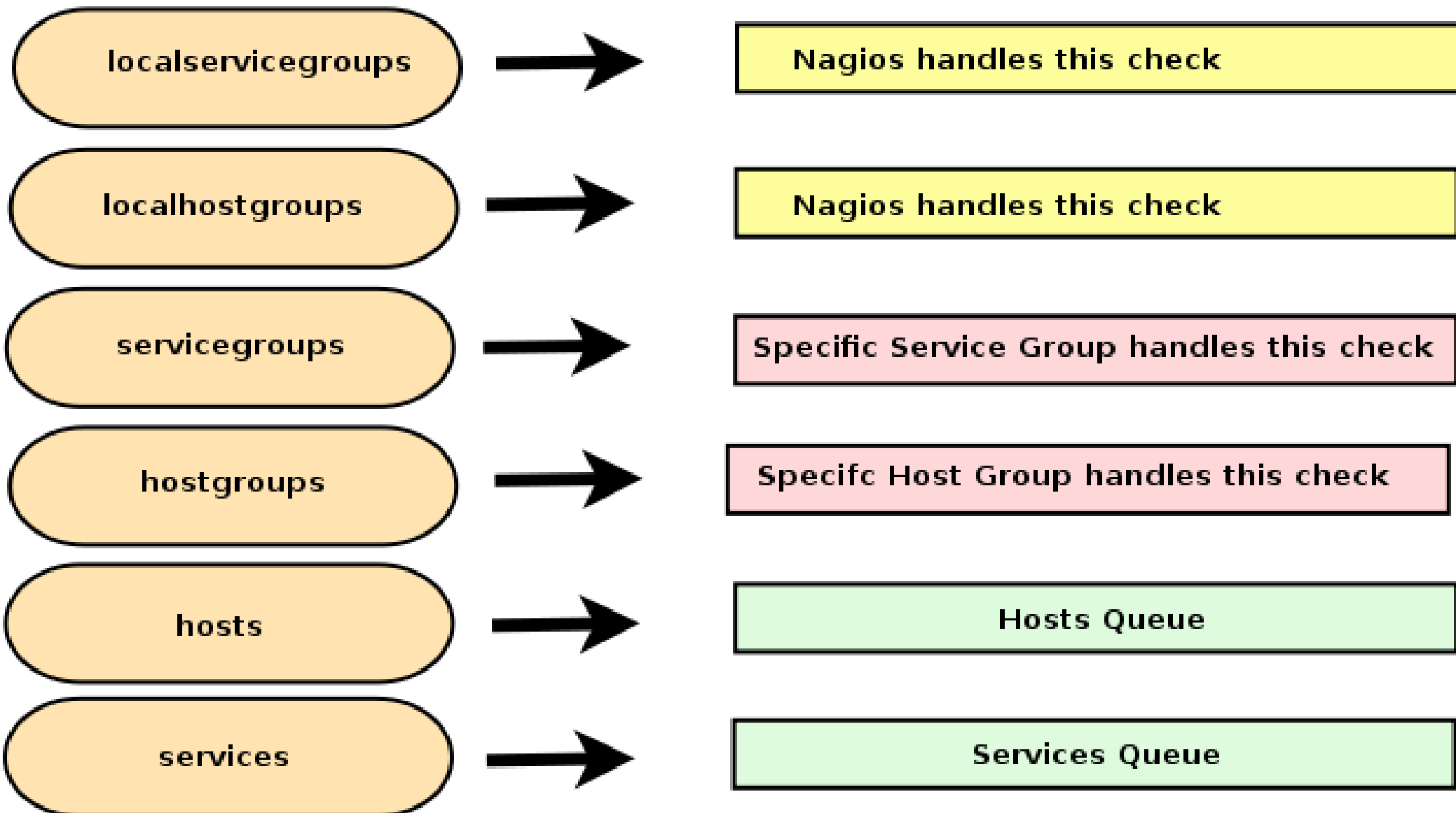
User Created Application :
NDOUtils
DNX
Mod-Gearman
MKLivestatus
Merlin

Mod-Gearman Advantages

▶ Reduce Overall Load on Nagios

▶ Move Problem Processes Off Nagios
Eventhandler waiting/hung
Plugin Delay – some plugins capture resources for up to 10 seconds

▶ Reduce Overhead when Forking Nagios Core
When Nagios has to fork a new process and the core is large, the system can take a major hit.

Mod-Gearman Queues

Queue Responsiblitiy

localservicegroups	→	Nagios handles this check
localhostgroups	→	Nagios handles this check
servicegroups	→	Specific Service Group handles this check
hostgroups	→	Specifc Host Group handles this check
hosts	→	Hosts Queue
services	→	Services Queue

Mod-Gearman Worker Configuration

▶ **Worker Identifier**
Unique identifier for worker, hostname

▶ **min-worker**
Minimum number of total workers

▶ **max-worker**
Maximum number of total workers

▶ **idle-timeout**
Time in seconds before idle worker exits

▶ **max-jobs**
Maximum number of jobs before worker exits

- ## Install Nagios Event Broker

  ```
  broker_module=/usr/local/lib/mod_gearman/mod_gearman.o
  config=/usr/local/etc/mod_gearman.conf
  ```

- ## Install Server: gearmand

  /etc/init.d/gearmand start

- ## Install Worker: mod_gearman_worker

  /etc/init.d/mod_gearman_worker start

- ## Configuration File

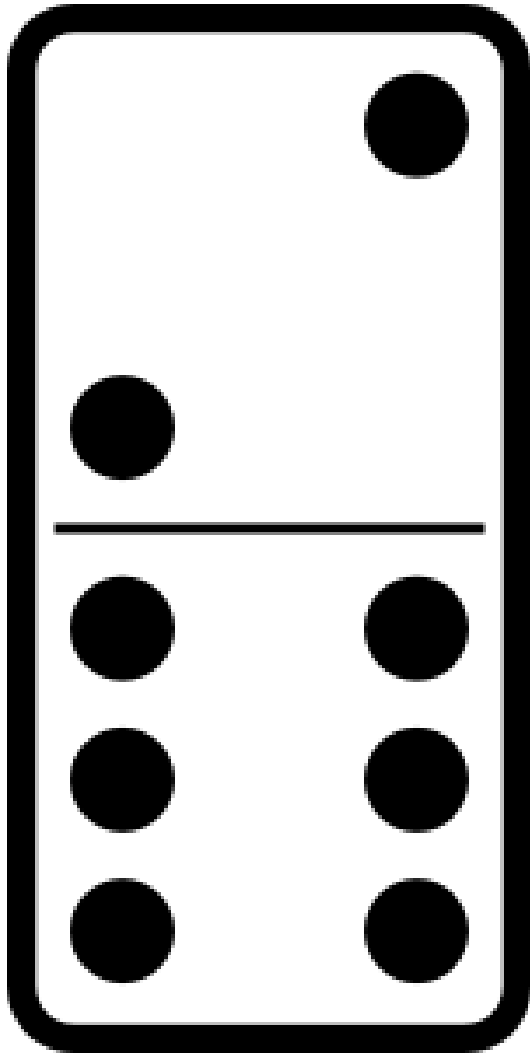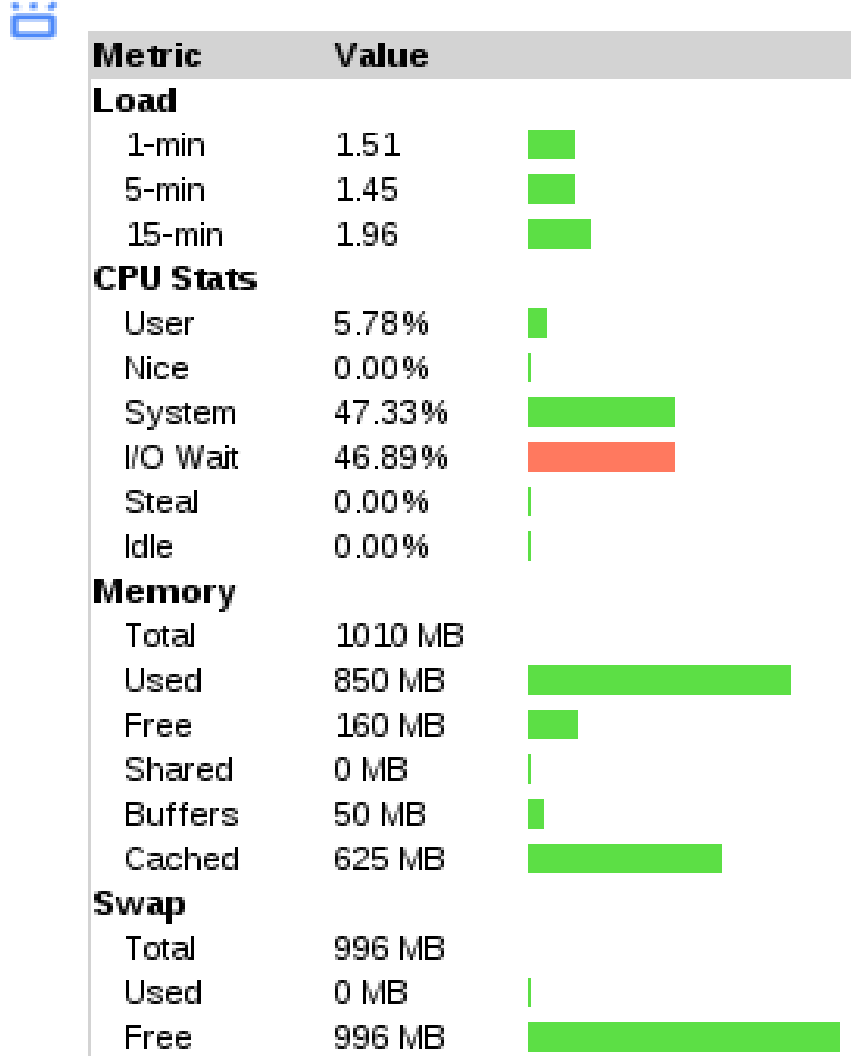 /usr/local/etc/mod_gearman.conf

Self-Load Balancing

Avoiding Falling Dominos

▶ Mod-Gearman Server

Monitoring and Distributing the Load

▶ Mod-Gearman Worker

Using a Nagios Event Broker to Lighten the Load
Using a Nagios Event Broker to Distribute the Load

▶ Reduce Latency

Latency reduced by up to 37%

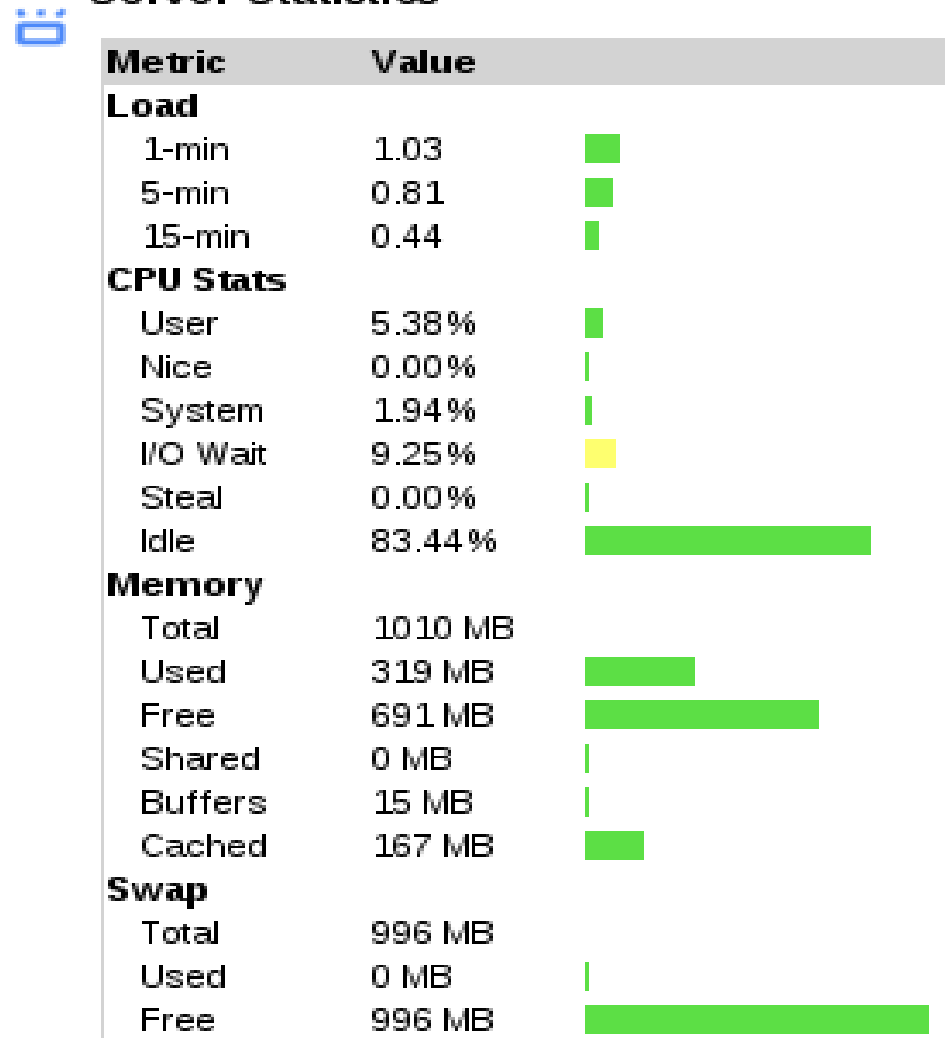▶ Extend Life in Crisis Events

▶ Preparation for Load Balancing

Self-Load Balancing: Before - After

Server Statistics

Metric	Value	
Load		
1-min	1.51	▮
5-min	1.45	▮
15-min	1.96	▮
CPU Stats		
User	5.78%	▮
Nice	0.00%	|
System	47.33%	▮▮▮
I/O Wait	46.89%	▮▮▮
Steal	0.00%	|
Idle	0.00%	|
Memory		
Total	1010 MB	
Used	850 MB	▮▮▮▮
Free	160 MB	▮
Shared	0 MB	|
Buffers	50 MB	▮
Cached	625 MB	▮▮▮
Swap		
Total	996 MB	
Used	0 MB	|
Free	996 MB	▮▮▮▮

Last Updated : 2011-09-04 04:22:33

Server Statistics

Metric	Value	
Load		
1-min	1.03	▮
5-min	0.81	▮
15-min	0.44	▮
CPU Stats		
User	5.38%	▮
Nice	0.00%	|
System	1.94%	▮
I/O Wait	9.25%	▮
Steal	0.00%	|
Idle	83.44%	▮▮▮▮▮
Memory		
Total	1010 MB	
Used	319 MB	▮▮
Free	691 MB	▮▮▮▮
Shared	0 MB	|
Buffers	15 MB	|
Cached	167 MB	▮
Swap		
Total	996 MB	
Used	0 MB	|
Free	996 MB	▮▮▮▮▮

Last Updated : 2011-09-04 04:53:55

Mod-Gearman: Server and Worker

```
PID     PPID    CPU  RAM
13480   1       0.0     1328    03:14:05 /usr/local/bin/mod_gearman_worker -d –
config=/usr/local/etc/mod_gearman.conf –
pidfile=/usr/local/var/mod_gearman/mod_gearman_worker.pid

13481  13480   0.0    1328    03:14:05 /usr/local/bin/mod_gearman_worker -d
--config=/usr/local/etc/mod_gearman.conf –
pidfile=/usr/local/var/mod_gearman/mod_gearman_worker.pid

19734   1       0.0     114580   01:07:50 /usr/local/sbin/gearmand -p 4730 -P
/usr/local/var/mod_gearman/gearmand.pid -d -j 0 -t 10 --log-
file=/usr/local/var/mod_gearman/gearmand.log -v -v --l
```

As you can see the worker uses very little in the way of resources (1328 KB of Memory) compared to the server which uses over 111 MB of Memory. When you look at these figures, you can see adding a worker to help shift the load on the local Nagios server is not a far fetched as you might originally think.

Load Balancing

Load Balancing with Mod-Gearman

NAGIOS SERVER

Workers Accept the Load
from the Nagios Server

host checks
service checks
eventhandlers
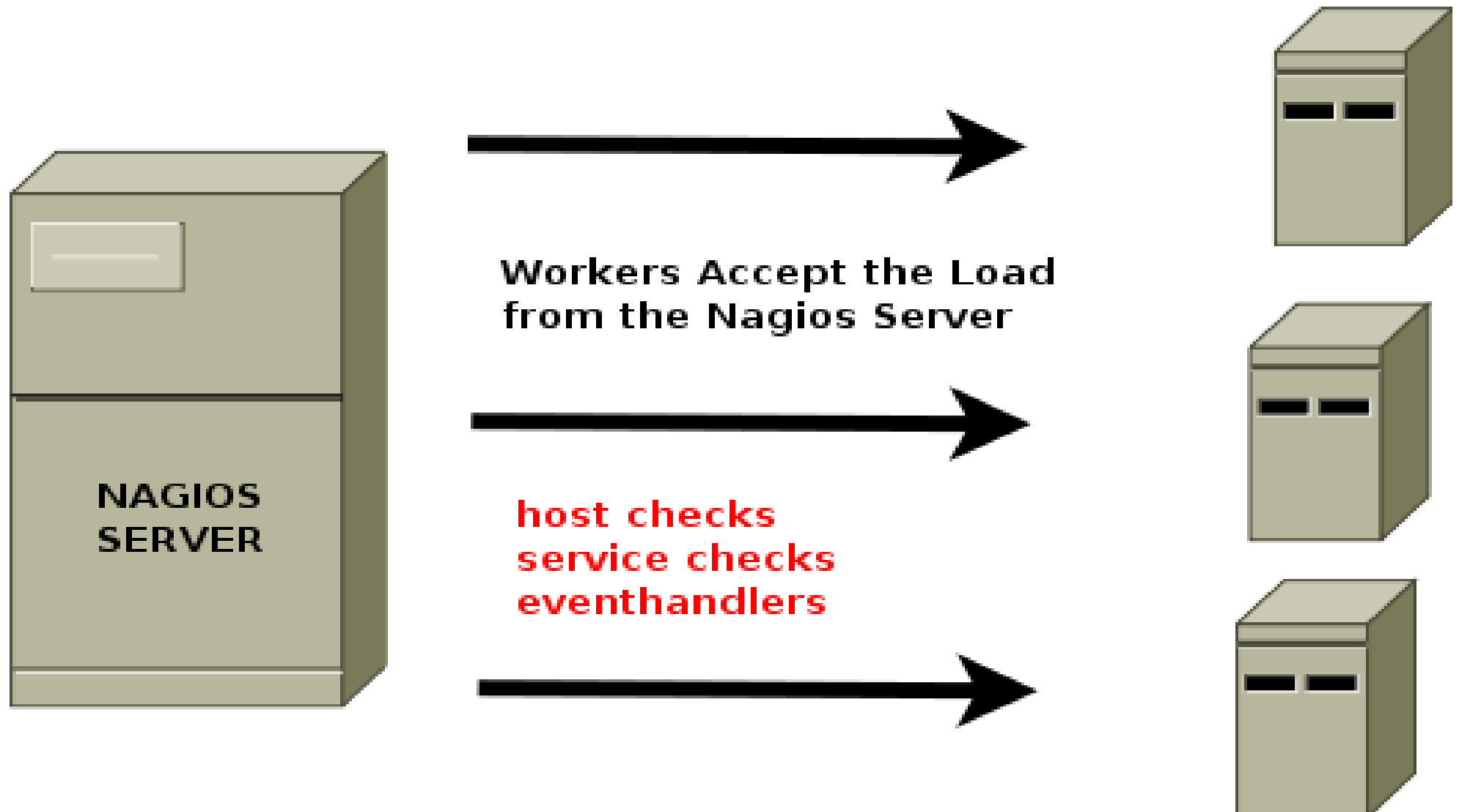
Load Balancing: Server and Worker

Server Configuration: /usr/local/etc/mod_gearman.conf

```
server=localhost:4730
eventhandler=yes
services=yes
hosts=yes
encryption=yes
key=linux23_Qg549K
```

Worker Configuration: /usr/local/etc/mod_gearman.conf

```
server=192.168.5.4:4730
eventhandler=yes
services=yes
hosts=yes
encryption=yes
key=linux23_Qg549K
```

Load Balancing: 2 Workers
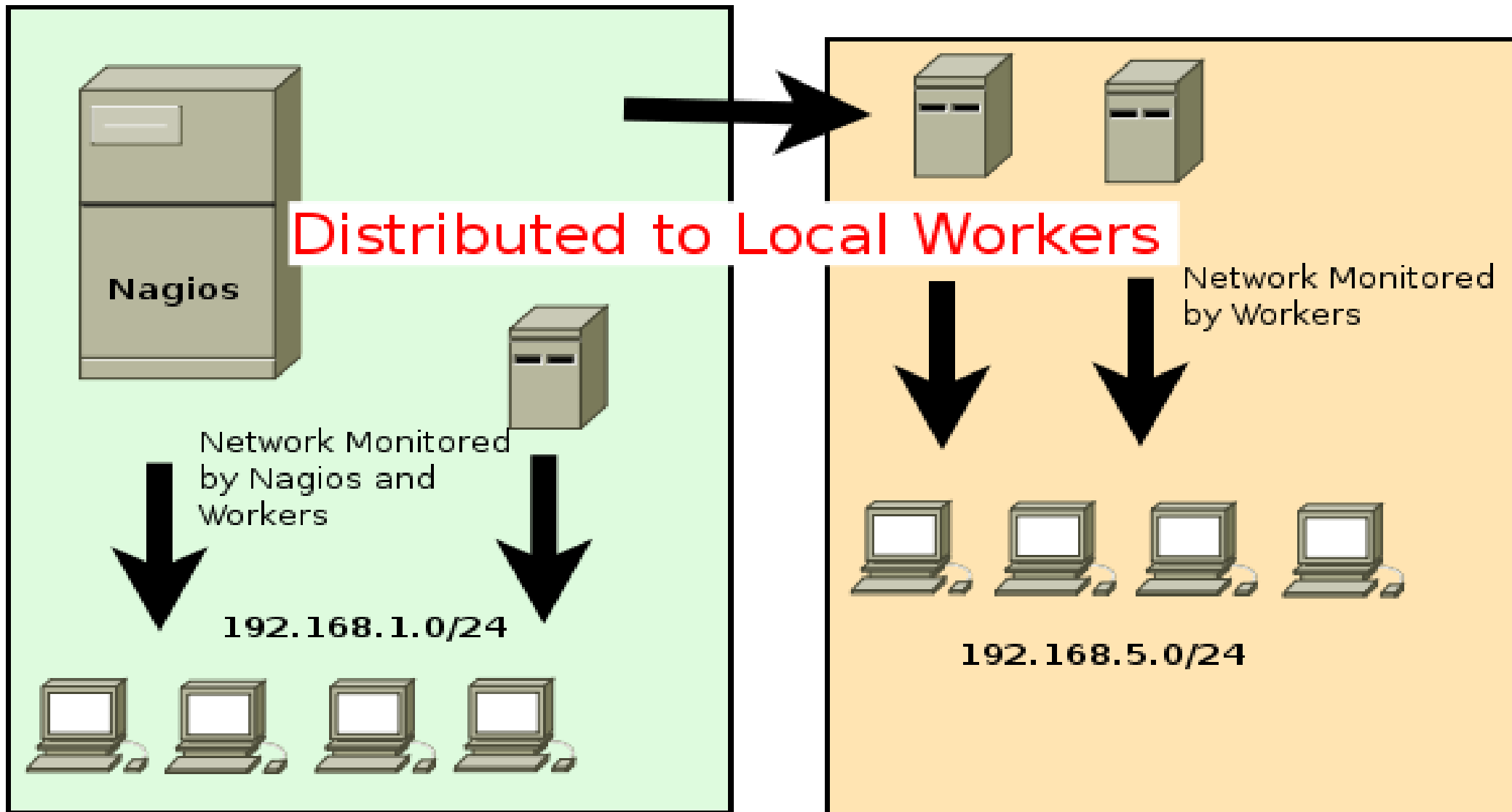
```
2011-09-04 23:41:38  -  localhost:4730  -  v0.14

 Queue Name                     | Worker Available | Jobs Waiting | Jobs Running
-------------------------------------------------------------------------------
 check_results                  |              1 |           0 |            0
 eventhandler                   |              2 |           0 |            0
 host                           |              2 |           0 |            0
 hostgroup_debian-servers       |              1 |           0 |            0
 service                        |              2 |           0 |            0
 worker_db                      |              1 |           0 |            0
 worker_localhost.localdomain   |              1 |           0 |            0
-------------------------------------------------------------------------------
```

Distributed to Local Workers

Nagios

Network Monitored by Nagios and Workers

192.168.1.0/24

Network Monitored by Workers

192.168.5.0/24

Distributed Monitoring with Mod-Gearman

Distributed Monitoring: Hostgroups

Server Configuration: /usr/local/etc/mod_gearman.conf

```
server=localhost:4730
eventhandler=yes
services=yes
hosts=yes
encryption=yes
key=linux23_Qg549K
```

Worker Configuration: /usr/local/etc/mod_gearman.conf

```
server=192.168.5.4:4730
eventhandler=yes
services=yes
hosts=yes
encryption=yes
key=linux23_Qg549K

hostgroups=debian-servers
```
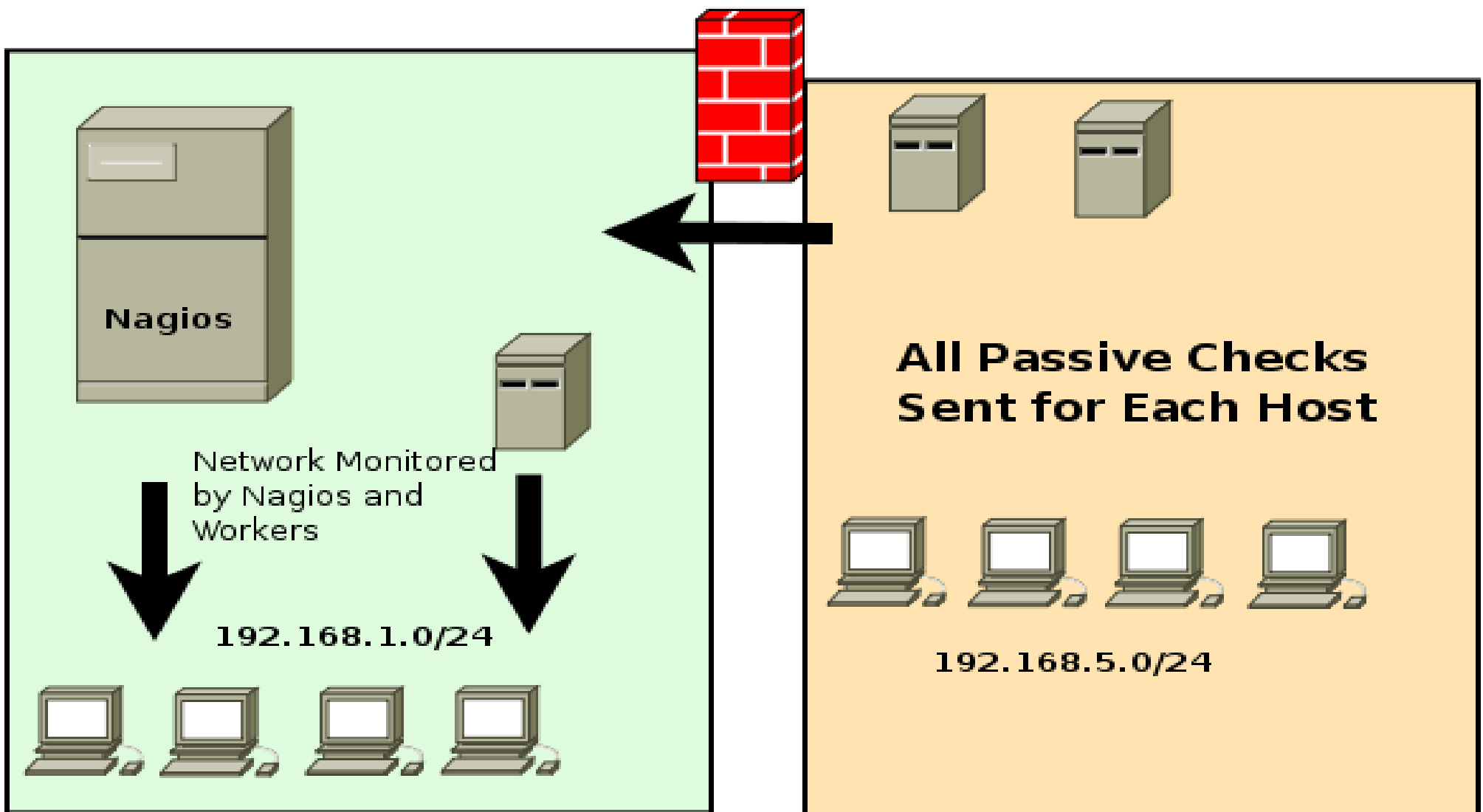
Passive Monitoring

Nagios

Network Monitored
by Nagios and
Workers

192.168.1.0/24

All Passive Checks
Sent for Each Host

192.168.5.0/24

Passive Checks with Mod-Gearman

Passive Monitoring: send_gearman

```bash
#!/bin/bash
# Mod-Gearman Passive User Check
users=`/usr/local/nagios/libexec/check_users -w 1 -c 5`
/usr/local/bin/send_gearman --server=192.168.5.4:4730 --encryption=yes
--key=linux23_Qg549K --host=db --service=GearUsers --message="$users"
--returncode=1




--server=192.168.5.4:4730
--encryption=yes
--key=linux23_Qg549K
--host=db
--service=GearUsers
--message="$users"
--returncode=1
```

Nagios
World Conference
North America

Passive Monitoring: Service Check

Service Status Details For All Hosts

Host	Service		Status	Last Check	Duration	Attempt	Status Information
db	GearUsers	?	WARNING	09-05-2011 13:26:04	???	1/3	USERS WARNING - 2 users currently logged in
	IMAP Port		OK	09-05-2011 12:50:04	0d 7h 58m 19s	1/3	TCP OK - 0.000 second response time on port 143
	POP3 Port		OK	09-05-2011 12:21:51	0d 7h 58m 12s	1/3	TCP OK - 0.000 second response time on port 110
	Postfix Port		OK	09-05-2011 12:00:17	0d 7h 58m 6s	1/3	TCP OK - 0.000 second response time on port 25
	SSH Port		OK	09-05-2011 11:35:21	0d 8h 43m 2s	1/3	TCP OK - 0.000 second response time on port 22
	Secure IMAPS		OK	09-05-2011 12:00:39	0d 7h 57m 44s	1/3	TCP OK - 0.000 second response time on port 993