openNMS®

Enterprise-Grade Open-Source Network Management

Large-Scale Linux Systems Monitoring with OpenNMS

Southern California Linux Expo, 26 Feb 2011

Jeff Gehlbach

jeffg@opennms.org

Who Am I; Why Am I Here?

First Linux distro: Slackware 3.1, 1994

First kernel from source: 1.2.13, 1995

Other FLOSS management apps: MRTG, 1998; Cricket; Cacti; Nagios

Proprietary management platforms: HP OpenView, 1999; Netcool OMNIbus; Empire SystemEDGE; Concord *e*Health

First SCaLE: 6x, 2008

Large-Scale

Highly subjective measure; hundreds +

Photo Credit:US National Oceanic and Atmospheric Administration

Linux

Major distributions of Linux 2.6

Also (with issues):

Mac OS X, *BSD, (Open)?Solaris 10+

Similar but different: AIX, HP-UX

Systems Monitoring

Really: Systems Management.

How to add them all to be managed?

Services up and responding quickly?

Something happened, how to know?

What's happening under the hood?

Something shiny to show the boss?

With OpenNMS®

World's First Enterprise-Grade Network Management Platform Developed Under the Open Source Model

Started in 1999 by ex-OpenView hackers

Maintained by the Order of the Green Polo

Supported, sponsored by my employer

Consistent model designed for huge scale

100% GPLv2 codebase

Will never suck

Will always be Free (as in Freedom)

Not Just for Systems

This talk focuses on managing servers.

OpenNMS manages much more.

→ Infrastructure (switches, routers, UPS)

→ Storage (SAN, NAS)

→ Environmental sensors, PDUs

→ Telco gear (TDM, 3G/GSM networks)

→ Anything with an IP address

→ Quite a few things without

Not “Based On X”

Built from the ground up

100% GPLv2 code base, Java

Makes extensive use of good libraries

Does not duct-tape in other apps

→ That way lies the end of scalability

→ Not to mention maintainability

Architectural decisions dictated by requirement to scale huge.

Use What Works For You

If you're happy, don't mess with it.

But maybe it wasn't designed for that...

Photo credit: Wikimedia Commons
Analogy: Alex Finger <af@genevainformation.ch>

Your Systems are Important

- The network exists to connect systems.
- Success → more systems all the time

OpenNMS is...

- Free
- Flexible
- Powerful
- Supported
- Designed to save you time
  - We consciously avoid design decisions that would put up scale walls, but HW matters
  - Properly sized hardware → awesome scale
  - Undersized hardware → EPIC FAIL

Sizing OpenNMS

Money-spending order:

→ Storage

→ Memory

→ CPU cores

→ More cores > faster cores

Sizing OpenNMS: Storage

Direct-attached

→ Many and fast spindles

→ Good SAS HBA w/ big BBWC

→ RAID5 is harmful. Seriously.

SAN

→ FC, iSCSI, NFS all fine

→ Mind the pathing

Sizing OpenNMS: Database

Give PostgreSQL its own server

Use PostgreSQL 8.4 or 9.0

If using < 8.4, you MUST tune it.

8.4+ tunes its own FSM. WIN.

Use the C-language IPLIKE sproc.

Images: pgforge; Wikimedia Commons; venganza.org

Sizing OpenNMS: Filesystems

1. Give PostgreSQL tablespaces and/or transaction logs own FS.

2. Give RRD files own FS.

2a. Separate perf, response

Each FS on its own I/O path, *i.e.* RAID volume, LUN.

Two RAID1 > One RAID10.

Sizing OpenNMS: Disk I/O

This is why disks matter :)

Enable store-by-group RRD persisting

```
# In opennms.properties
org.opennms.rrd.storeByGroup=true
```

Use MNIO JRobin back-end

→ In OpenNMS 1.8.10+

```
# In rrd-configuration.properties
org.jrobin.core.RrdBackendFactory=MNIO
```

Sizing OpenNMS: Memory

For large-scale, start at 4GB

64-bit kernel so you can use it!

Give PostgreSQL plenty.

Give the JVM plenty.

→ Default 256MB heap too small

→ Max PermGen size also

Mapping Needs to Capabilities

How to add nodes to be managed?

→Discovery and Provisioning

Services up and responding quickly?

→Service Monitoring (polling)

Something happened, how to know?

→Event Management and Notifications

What's happening under the hood?

→Performance Data Collection

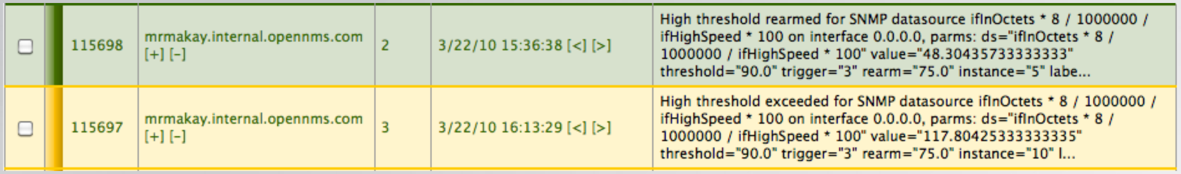
Performance Data Thresholds

Basic threshold: single variable

Expression-based: multiple variables

Configurable time-over-threshold trigger

Evaluate:



□	115698	mrmakay.internal.opennms.com [+] [-]	2	3/22/10 15:36:38 [<] [>]	High threshold rearmed for SNMP datasource ifInOctets * 8 / 1000000 / ifHighSpeed * 100 on interface 0.0.0.0, parms: ds="ifInOctets * 8 / 1000000 / ifHighSpeed * 100" value="48.30435733333333" threshold="90.0" trigger="3" rearm="75.0" instance="5" labe...
□	115697	mrmakay.internal.opennms.com [+] [-]	3	3/22/10 16:13:29 [<] [>]	High threshold exceeded for SNMP datasource ifInOctets * 8 / 1000000 / ifHighSpeed * 100 on interface 0.0.0.0, parms: ds="ifInOctets * 8 / 1000000 / ifHighSpeed * 100" value="117.80425333333335" threshold="90.0" trigger="3" rearm="75.0" instance="10" l...

→ High / low (optional re-arm)

→ Relative change (ratio, no re-arm)

→ Absolute change (optional re-arm)

Data Collection Protocols

The great thing about standards...

SNMP: Standards-track, tunable, robust

→ **Use SNMP when you can.**

HTTP: Handy when SNMP is impractical

JMX: Peek inside JVM, app containers

NSClient: Flaky at scale; can be handy

WMI: Outside scope of this talk :)

XMP: Next-generation P2P protocol

Discovery and Provisioning

Discovery: Awareness of a previously unknown IP address, usually via ping

Image: Wikimedia Commons

Image: Wikimedia Commons

Provisioning: Finding out all we can and representing results in our model.

Service(s) → Interface(s) → Node

Provisioning

Capsd: Legacy capabilities scanner.

Automatic Provisioning: Seed an IP address; scan for interfaces and services.

Directed Provisioning: Seed an exact set of known IP interfaces and services.

Policy-Based Provisioning: Seed an IP address; scan for interfaces and services, deciding on persistence, data collection, service monitoring, categorization...

Provisioning (cont'd)

External provisioning sources too!

DNS import: Do a zone transfer, create nodes and interfaces from 'A' records

Your DB: Write a CGI that generates XML describing your systems, feed the URL to Provisiond, watch the magic happen

EC2-compatible APIs: In a feature branch, track me down if you want to talk

Service Monitoring

Is a service on an interface responding?

Simple: ICMP Ping, HTTP GET

Moderate: Processes via SNMP

Advanced: Page Sequence,

Mail Transport

Optionally store response times

Event Management

Something happened in the network...

Internal: A service was found to be down

External: SNMP traps, syslog, TL1

Custom: XML-formatted events over TCP

Events optionally “de-duplicated” to alarms with a “count” attribute.

Notification Management

...now tell one or more people about it.

E-Mail: JavaMail API. Avoid `/bin/mail`.

XMPP: To individuals or group chat.

Asterisk: “Press 1 to acknowledge...”

Custom: Fork a command. Use sparingly.

Reusable destination paths, escalations, auto-acknowledgement for certain cases.

Performance Data Collection

Peer inside the system to find out...

Network: Traffic, discards, errors, *cast...

CPU/Memory: Utilization, time-in-state...

Filesystem/Disk: Utilization, reads, writes

Derived: Load average, users, processes

Whatever: Straightforwardly extensible

Store data for graphing, TopN reporting

Threshold data in real-time (→ events)

Net-SNMP

Multiplies the power of OpenNMS.

Ubiquitous: **The** FLOSS UNIX SNMP agent

Capable: Many useful MIBs built in

Extensible: Glue in arbitrary commands

Autonomous: Self-monitor, send traps

Net-SNMP self-monitoring

Log Files: `logmatch, file`

Processes: `proc, procfix`

System: `load, swap`

Whatever: Net-SNMP can self-monitor any MIB object right on the box.

Send DISMAN traps: `iquerySecName, trap2sink, monitor`

(→ Events, alarms, notifications)

Extending Net-SNMP

Old Directives: `sh`, `exec` (don't use)

New Directives: `extend`, `pass`

Run a command, glue its output into the MIB tree – with configurable caching

Example: Provide missing CPU count

```
 extend .1.3.6.1.4.1.5813.255 cpuCount /bin/egrep -c \
  '^processor.*?:.*?[0-9]+' /proc/cpuinfo
```

Collect this new object in OpenNMS:

```
<mibObj oid=”.1.3.6.1.4.1.5813.255...100” instance=”1”

 alias=”nsExtCpuCount” />
```

Net-SNMP Issues

Old Releases: 64-bit problems abound

5.2.1.2: Nasty interface counter problem

Release <5.5: `dskTable` w/FS >= 2TB :'(

Annoyance: No CPU core count object

Image Credit: teh internets (sadly, unknown)

Anti-patterns

Image Credit: teh intarweb

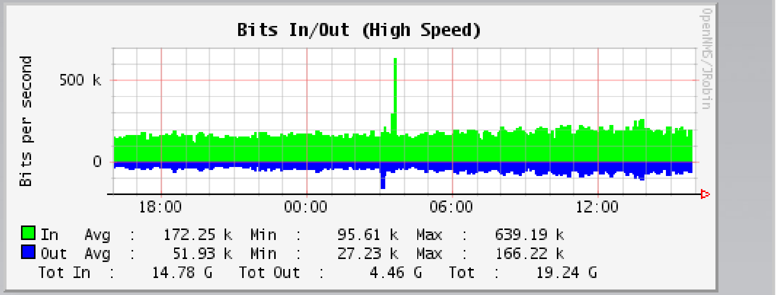
Anti-patterns

- Forking a lot of stuff on the NMS box (notification commands, `GpMonitor`)
- Using SSH + `$SHELL` as an agent
- Favoring TCP-based protocols because SNMP / UDP is “unreliable”
- Keeping RRD performance data forever
- Keeping RDBMS event data forever
- Styling your OpenNMS configs after your Nagios configs

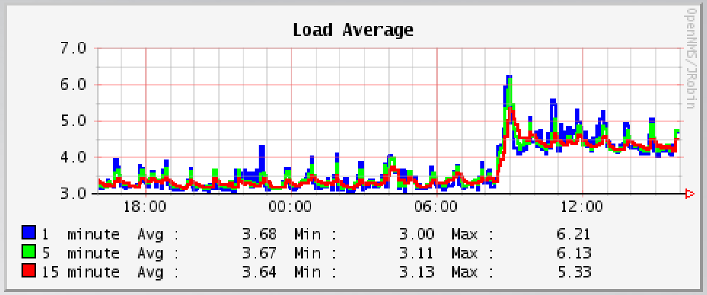
Something Shiny for the Boss

- Resource graphs for visualization of response and performance data
- PDF availability reports (month, YTD)
- JasperReports integration
  - Events, alarms, etc. from RDBMS
  - Response and performance data from RRD files
  - Generated ad-hoc or scheduled; e-mail

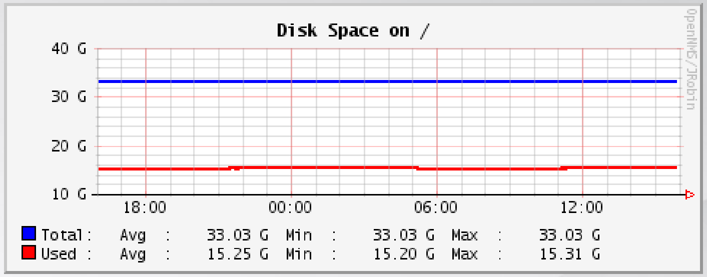
Performance Data Eye Candy



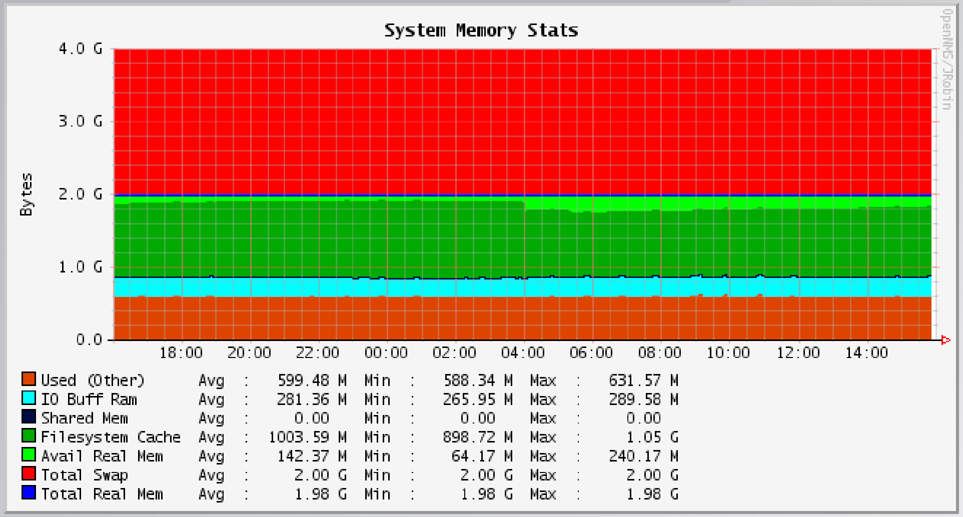
```
🟩 In   Avg  :   172.25 k  Min  :   95.61 k  Max  :   639.19 k
🟦 Out  Avg  :    51.93 k  Min  :   27.23 k  Max  :   166.22 k
  Tot In  :   14.78 G   Tot Out  :    4.46 G   Tot  :   19.24 G
```



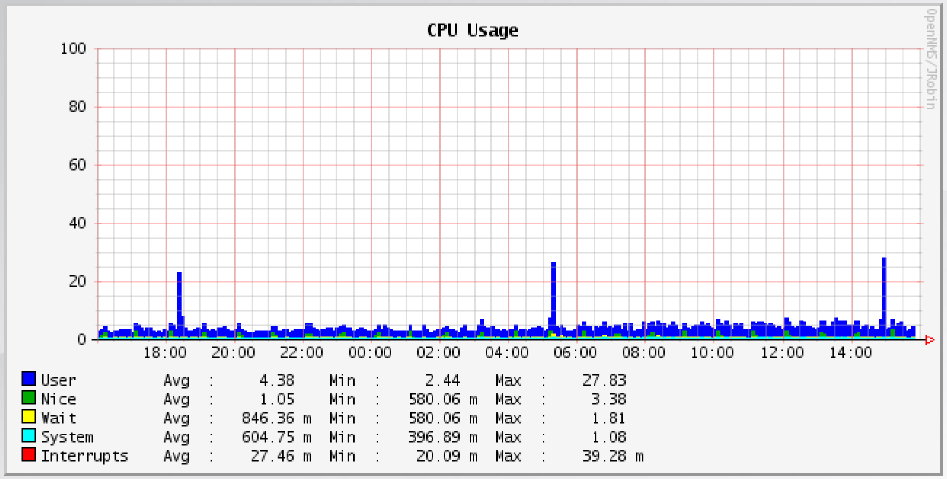
```
🟦 1  minute  Avg :    3.68  Min :    3.00  Max :    6.21
🟩 5  minute  Avg :    3.67  Min :    3.11  Max :    6.13
🟥 15 minute  Avg :    3.64  Min :    3.13  Max :    5.33
```



```
🟦 Total:  Avg  :  33.03 G  Min  :  33.03 G  Max  :  33.03 G
🟥 Used :  Avg  :  15.25 G  Min  :  15.20 G  Max  :  15.31 G
```

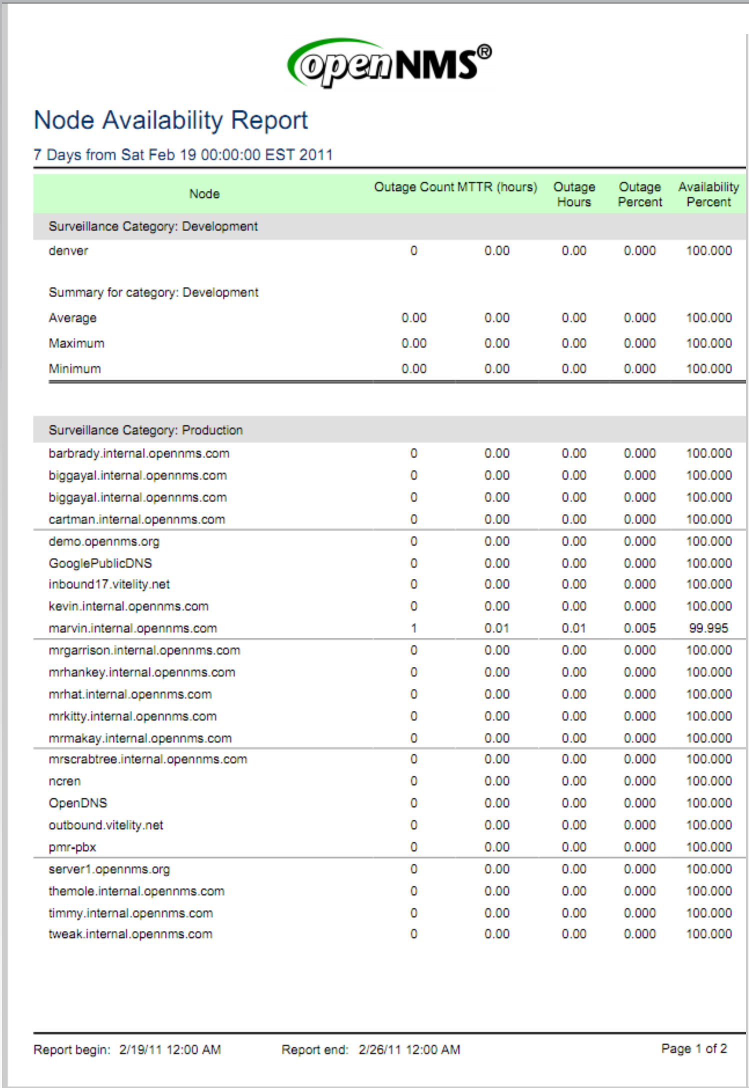


```
🟧 Used (Other)       Avg  :   599.48 M  Min  :   588.34 M  Max  :   631.57 M
🟦 IO Buff Ram        Avg  :   281.36 M  Min  :   265.95 M  Max  :   289.58 M
⬛ Shared Mem         Avg  :     0.00    Min  :     0.00    Max  :     0.00
🟩 Filesystem Cache   Avg  :  1003.59 M  Min  :   898.72 M  Max  :     1.05 G
🟩 Avail Real Mem     Avg  :   142.37 M  Min  :    64.17 M  Max  :   240.17 M
🟥 Total Swap         Avg  :     2.00 G  Min  :     2.00 G  Max  :     2.00 G
🟦 Total Real Mem     Avg  :     1.98 G  Min  :     1.98 G  Max  :     1.98 G
```



```
🟦 User        Avg  :    4.38    Min  :    2.44    Max  :   27.83
🟩 Nice        Avg  :    1.05    Min  :  580.06 m  Max  :    3.38
🟨 Wait        Avg  :  846.36 m  Min  :  580.06 m  Max  :    1.81
🟦 System      Avg  :  604.75 m  Min  :  396.89 m  Max  :    1.08
🟥 Interrupts  Avg  :   27.46 m  Min  :   20.09 m  Max  :   39.28 m
```

JasperReports Eye Candy

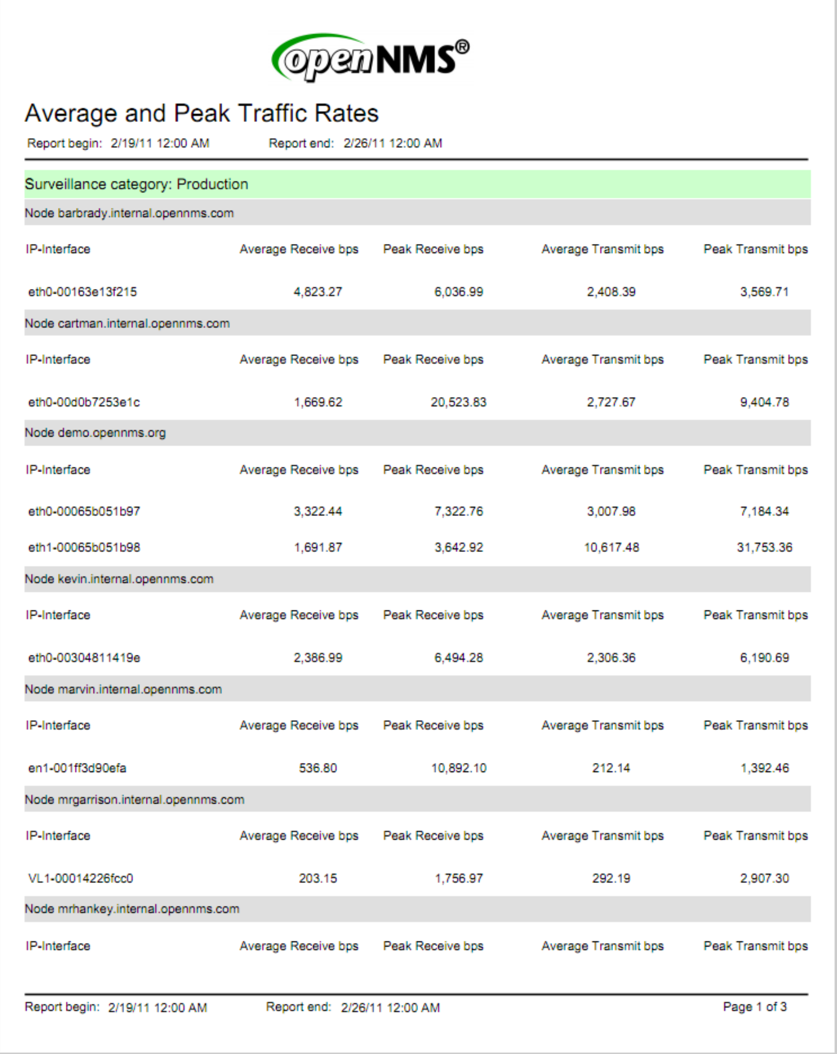


Node Availability Report

7 Days from Sat Feb 19 00:00:00 EST 2011

Node	Outage Count	MTTR (hours)	Outage Hours	Outage Percent	Availability Percent
Surveillance Category: Development					
denver	0	0.00	0.00	0.000	100.000
Summary for category: Development					
Average	0.00	0.00	0.00	0.000	100.000
Maximum	0.00	0.00	0.00	0.000	100.000
Minimum	0.00	0.00	0.00	0.000	100.000
Surveillance Category: Production					
barbrady.internal.opennms.com	0	0.00	0.00	0.000	100.000
biggayal.internal.opennms.com	0	0.00	0.00	0.000	100.000
biggayal.internal.opennms.com	0	0.00	0.00	0.000	100.000
cartman.internal.opennms.com	0	0.00	0.00	0.000	100.000
demo.opennms.org	0	0.00	0.00	0.000	100.000
GooglePublicDNS	0	0.00	0.00	0.000	100.000
inbound17.vitelity.net	0	0.00	0.00	0.000	100.000
kevin.internal.opennms.com	0	0.00	0.00	0.000	100.000
marvin.internal.opennms.com	1	0.01	0.01	0.005	99.995
mrgarrison.internal.opennms.com	0	0.00	0.00	0.000	100.000
mrhankey.internal.opennms.com	0	0.00	0.00	0.000	100.000
mrhat.internal.opennms.com	0	0.00	0.00	0.000	100.000
mrkitty.internal.opennms.com	0	0.00	0.00	0.000	100.000
mrmakay.internal.opennms.com	0	0.00	0.00	0.000	100.000
mrscrabtree.internal.opennms.com	0	0.00	0.00	0.000	100.000
ncren	0	0.00	0.00	0.000	100.000
OpenDNS	0	0.00	0.00	0.000	100.000
outbound.vitelity.net	0	0.00	0.00	0.000	100.000
pmr-pbx	0	0.00	0.00	0.000	100.000
server1.opennms.org	0	0.00	0.00	0.000	100.000
themole.internal.opennms.com	0	0.00	0.00	0.000	100.000
timmy.internal.opennms.com	0	0.00	0.00	0.000	100.000
tweak.internal.opennms.com	0	0.00	0.00	0.000	100.000

Report begin: 2/19/11 12:00 AM   Report end: 2/26/11 12:00 AM   Page 1 of 2



Average and Peak Traffic Rates

Report begin: 2/19/11 12:00 AM   Report end: 2/26/11 12:00 AM

Surveillance category: Production

Node barbrady.internal.opennms.com

IP-Interface	Average Receive bps	Peak Receive bps	Average Transmit bps	Peak Transmit bps
eth0-00163e13f215	4,823.27	6,036.99	2,408.39	3,569.71

Node cartman.internal.opennms.com

IP-Interface	Average Receive bps	Peak Receive bps	Average Transmit bps	Peak Transmit bps
eth0-00d0b7253e1c	1,669.62	20,523.83	2,727.67	9,404.78

Node demo.opennms.org

IP-Interface	Average Receive bps	Peak Receive bps	Average Transmit bps	Peak Transmit bps
eth0-00065b051b97	3,322.44	7,322.76	3,007.98	7,184.34
eth1-00065b051b98	1,691.87	3,642.92	10,617.48	31,753.36

Node kevin.internal.opennms.com

IP-Interface	Average Receive bps	Peak Receive bps	Average Transmit bps	Peak Transmit bps
eth0-00304811419e	2,386.99	6,494.28	2,306.36	6,190.69

Node marvin.internal.opennms.com

IP-Interface	Average Receive bps	Peak Receive bps	Average Transmit bps	Peak Transmit bps
en1-001ff3d90efa	536.80	10,892.10	212.14	1,392.46

Node mrgarrison.internal.opennms.com

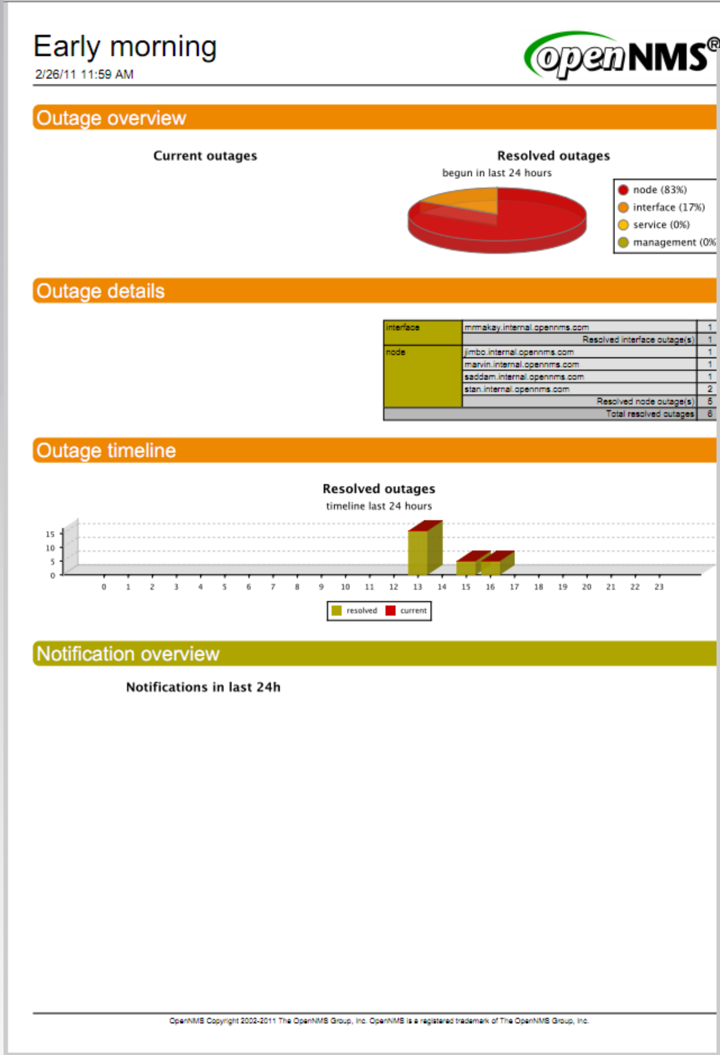
IP-Interface	Average Receive bps	Peak Receive bps	Average Transmit bps	Peak Transmit bps
VL1-00014226fcc0	203.15	1,756.97	292.19	2,907.30

Node mrhankey.internal.opennms.com

IP-Interface	Average Receive bps	Peak Receive bps	Average Transmit bps	Peak Transmit bps

Report begin: 2/19/11 12:00 AM   Report end: 2/26/11 12:00 AM   Page 1 of 3

MOAR!1! JasperReports Eye Candy



Early morning

2/26/11 11:59 AM

Outage overview

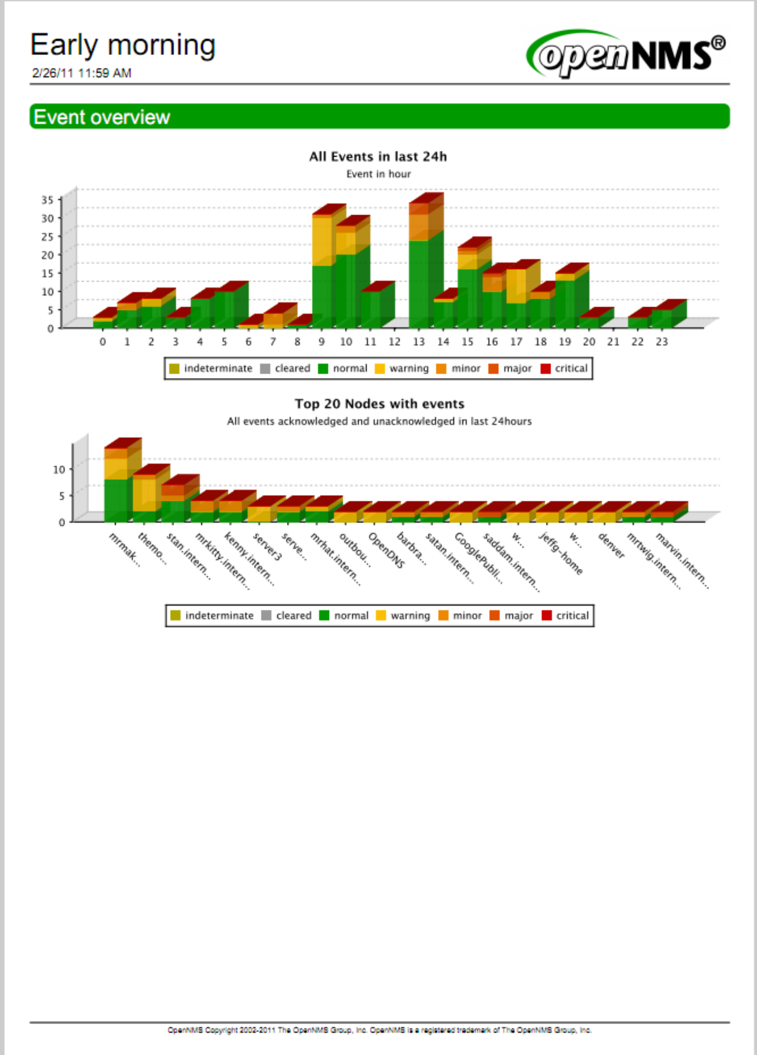
Outage details

interface	mrmakay.internal.opennms.com	1
	Resolved interface outage(s)	1
node	jimbo.internal.opennms.com	1
	marvin.internal.opennms.com	1
	saddam.internal.opennms.com	1
	stan.internal.opennms.com	2
	Resolved node outage(s)	5
Total resolved outages		6

Outage timeline

Notification overview

Notifications in last 24h



Early morning

2/26/11 11:59 AM

Event overview

Questions, Contact

Ask away!

identi.ca: @jeffg (group !opennms)

E-mail: jeffg@opennms.org

IRC (Freenode): jeffg – #opennms

Twitter: @jgehlbach

License