

Trending with Purpose

Jason Dixon



Monitoring

- Nagios
 - Fault Detection
 - Notifications
 - Escalations
 - Acknowledgements/Downtime
 - <http://www.nagios.org/>

Nagios

General

- Home
- Documentation

Current Status

- Tactical Overview
- Map
- Hosts
- Services
- Host Groups

- Summary
- Grid

Service Groups

- Summary
- Grid

Problems

- Services (Unhandled)
- Hosts (Unhandled)
- Network Outages

Quick Search:

Reports

- Availability
- Trends
- Alerts

- History
- Summary
- Histogram

- Notifications
- Event Log

System

- Comments
- Downtime
- Process Info
- Performance Info

Current Network Status

Last Updated: Fri Mar 18 03:20:42 UTC 2011
 Updated every 30 seconds
 Nagios® Core™ 3.2.1 - www.nagios.org
 Logged in as jolixon

- [View History For all hosts](#)
- [View Notifications For All Hosts](#)
- [View Host Status Detail For All Hosts](#)

Display Filters:

Host Status Types: All
 Host Properties: Any
 Service Status Types: All Problems
 Service Properties: Any

Host Status Totals

Up	Down	Unreachable	Pending
176	4	0	0
All Problems		All Types	
4		100	

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
857	12	0	13	0
All Problems		All Types		
25		882		

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ibexon	Backup Health	CRITICAL	2011-03-18 03:12:47	2d 23h 17m 55s	1/5	(Service Check Timed Out)
	Disk 1	CRITICAL	2011-03-18 03:16:47	2d 23h 13m 55s	1/5	CHECK_NRPE: Socket timeout after 30 seconds.
	Disk 2	CRITICAL	2011-03-18 03:16:47	2d 23h 13m 55s	1/5	CHECK_NRPE: Socket timeout after 30 seconds.
	Load	CRITICAL	2011-03-18 03:16:49	2d 23h 13m 54s	1/5	CHECK_NRPE: Socket timeout after 30 seconds.
	PING	CRITICAL	2011-03-18 03:18:49	2d 23h 11m 54s	1/5	PING CRITICAL - Packet loss = 100%
	Raid Health	CRITICAL	2011-03-18 03:16:47	2d 23h 13m 55s	1/5	CHECK_NRPE: Socket timeout after 30 seconds.
	SSH	CRITICAL	2011-03-18 03:16:47	2d 23h 13m 55s	1/5	CRITICAL - Socket timeout after 30 seconds
	Services	CRITICAL	2011-03-18 03:16:48	2d 23h 13m 54s	1/5	CHECK_NRPE: Socket timeout after 30 seconds.
	Swap Usage	CRITICAL	2011-03-18 03:16:49	2d 23h 13m 54s	1/5	CHECK_NRPE: Socket timeout after 30 seconds.
	Total Processes	CRITICAL	2011-03-18 03:16:48	2d 23h 13m 54s	1/5	CHECK_NRPE: Socket timeout after 30 seconds.
	UPS	CRITICAL	2011-03-18 03:16:47	0d 6h 13m 55s	1/5	CHECK_NRPE: Socket timeout after 30 seconds.
	UPS0.Process	CRITICAL	2011-03-18 03:16:49	2d 23h 13m 54s	1/5	CHECK_NRPE: Socket timeout after 30 seconds.
	octopus	OS Updates	WARNING	2011-03-18 03:08:36	10d 2h 21m 29s	5/5
zfs	OS Updates	WARNING	2011-03-18 03:09:13	10d 2h 8m 10s	5/5	Updates needed: coreutils.x86_64 cups.x86_64 cups-libs.x86_64 curl.x86_64 device-mapper-multipath.x86_64 gdb.x86_64 git.x86_64 git-email.x86_64 git-gui.x86_64 gtk.x86_64 kernel.x86_64 kernel-devel.x86_64 kernel-headers.x86_64 kpartx.x86_64 krb5-libs.i386 krb5-libs.x86_64 krb5-workstation.x86_64 ksh.x86_64 libmbclient.x86_64 libffi.x86_64 lm2.x86_64 mysql.x86_64 mysql-server.x86_64 nginx.x86_64 openldap.i386 openldap.x86_64 openldap-clients.x86_64 pango.x86_64 perl.x86_64 perl-Git.x86_64 postgresql-libs.i386 postgresql-libs.x86_64 postgresql94.x86_64 postgresql94-devel.x86_64 postgresql94-libs.x86_64 postgresql94-piperl.x86_64 postgresql94-server.x86_64 python26.x86_64 python26-libs.x86_64 samba.x86_64 samba-client.x86_64 samba-common.x86_64 subversion.i386 subversion.x86_64 vnc.x86_64 vsftpd.x86_64 xulrunner.x86_64 yum-rh-plugin.noarch

Nagios

- Pros
 - Free
 - Extensible
 - Plugins
 - Configuration templates
 - Popular (lesser of all free evils)
 - Log metrics (“performance data”)

Nagios

- Cons
 - Interface
 - (Lack of) Scalability
 - Promotes bad habits
 - Acknowledgements never expire
 - Configuration (over-)flexibility
 - Flapping

Nagios Demonstration

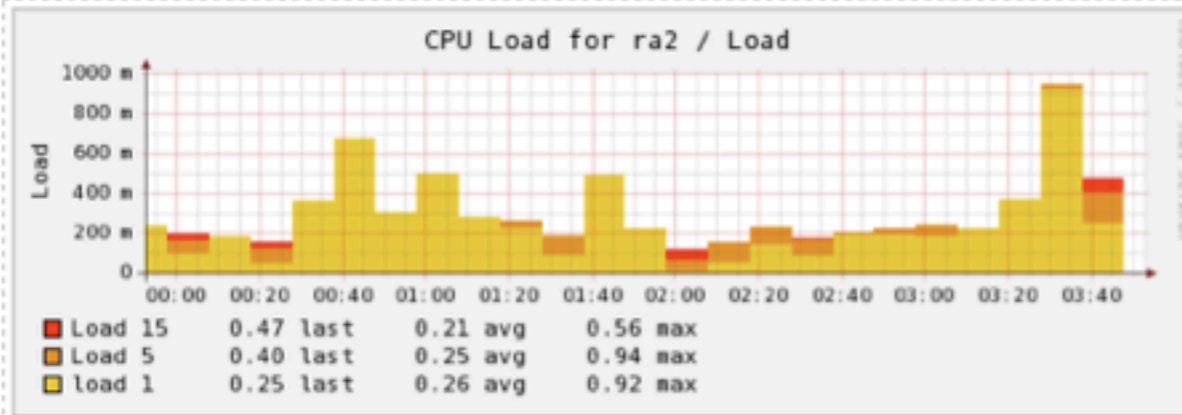
[URL redacted]

Trending

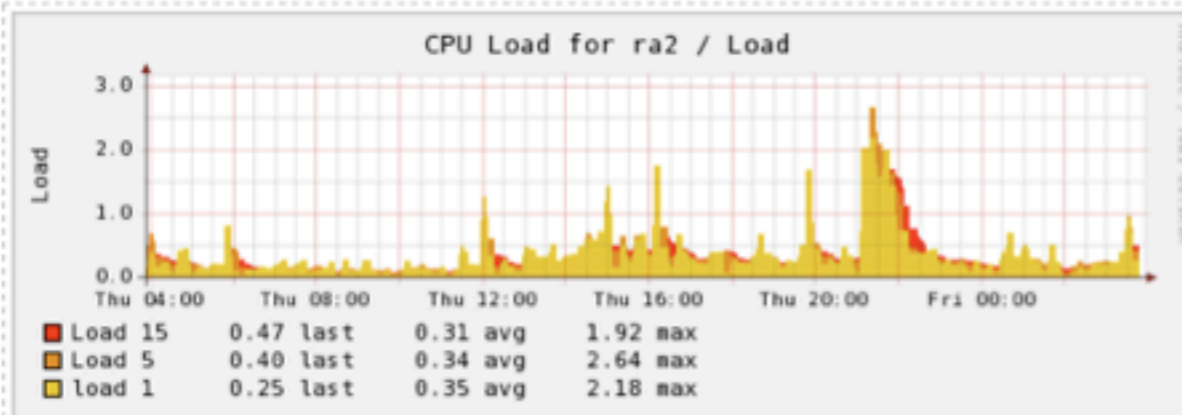
- PNP4Nagios
 - Retrieves Nagios performance data
 - Creates graphs with RRD
 - Basic dashboard capabilities
 - Limited introspection/correlation
 - <http://www.pnp4nagios.org/>

Service Overview

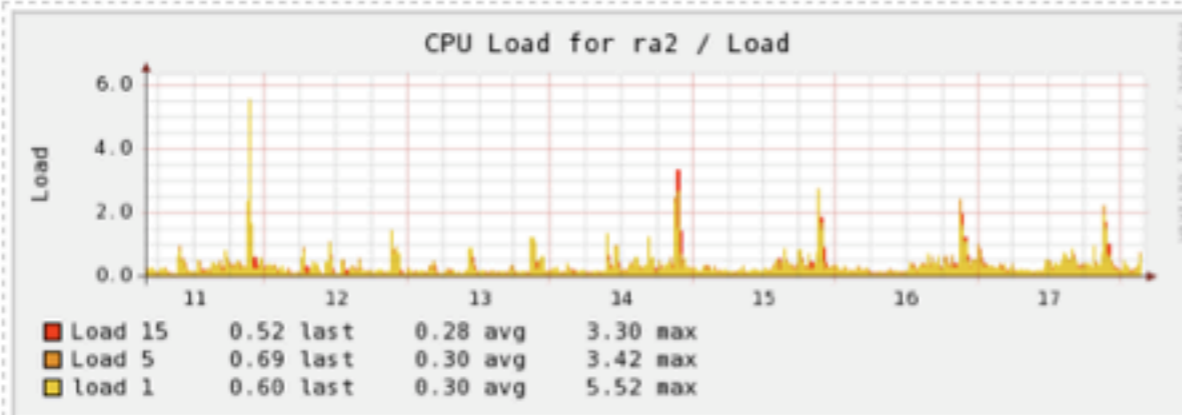
4 Hours (17.03.11 23:52 - 18.03.11 3:52)



24 Hours (17.03.11 3:52 - 18.03.11 3:52)



One Week (11.03.11 3:52 - 18.03.11 3:52)



Search:



Host: [ra2](#)
Service: [Load](#)
Hoststate: UP [HARD]
Servicestate: OK [HARD]
Created: 18.03.11 3:48

- [Host Perdata](#)
- [Current_Users](#)
- [Disk_1](#)
- [Disk_2](#)
- [Load](#)
- [Mail_Queue](#)
- [PING](#)
- [Swap_Usage](#)

PNP4Nagios Demonstration

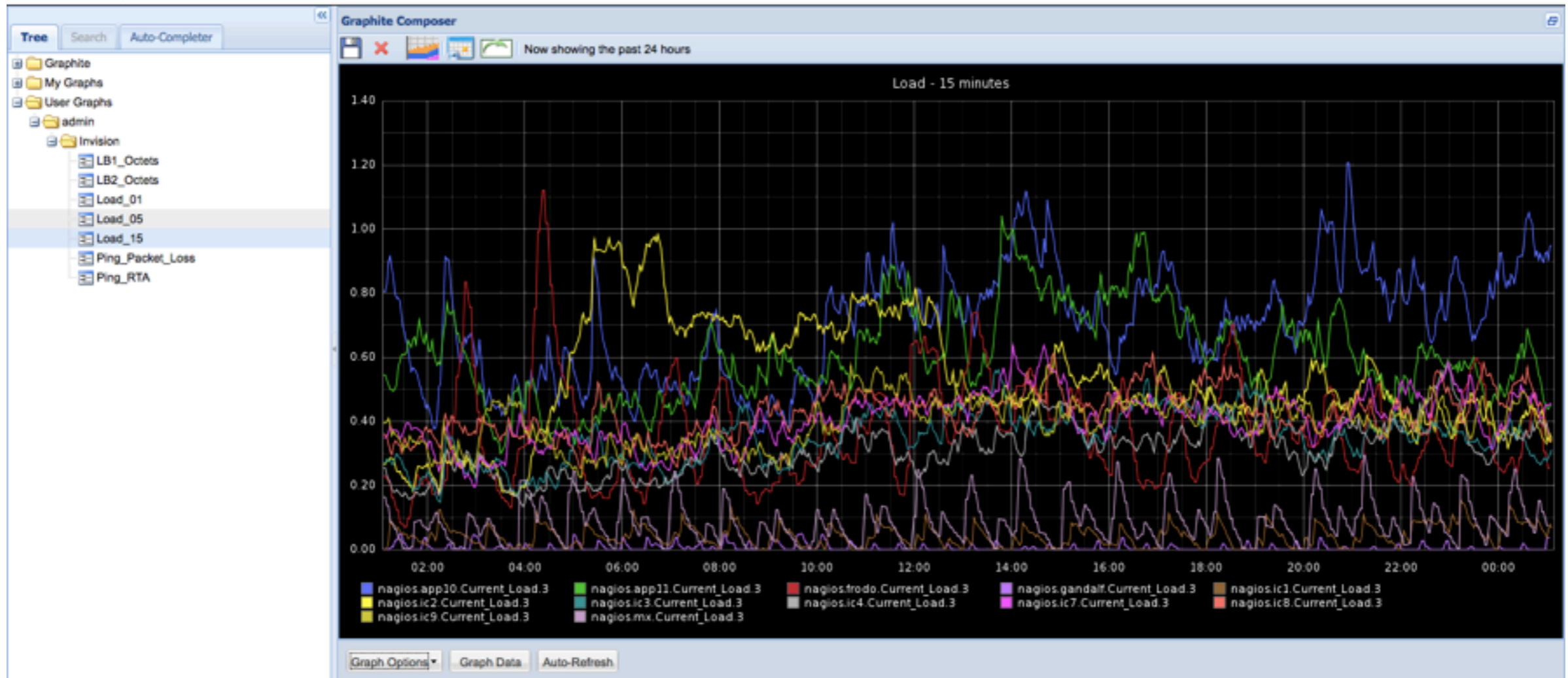
[URL redacted]

Advanced Graphing

- Graphite
 - Metric storage
 - Complex graph creation
 - Web and “CLI” interfaces
 - Created and released by Orbitz.com
 - <http://graphite.wikidot.com/>

graphite

[Command-Line Interface](#)
Logged in as admin, [logout](#) ([edit profile](#))
[Documentation](#)



Graphite

- Pros
 - Horizontally scalable
 - Rapid graph prototyping (CLI)
 - Graph disparate data points
 - Numerous formulas available
 - derive, transform, average, sum, etc...
 - Share graphs with other users
 - Supports existing RRD databases

Graphite

- Cons
 - Not a dashboard
 - No hover details

Graphite Components

- Carbon storage engine
 - agent - starts other daemons, receives metrics and pipelines them to cache
 - cache - caches metrics for real-time graphing, pipelines them to persister
 - persister - writes persistent data to disk

Graphite Components

- Whisper - metrics database format
 - Supplanted RRDtool
 - Accepts out-of-order data
 - Supports pipelining of data in a single operation (multiplexing)

Graphite Components

- Graphite
 - Traditional web interface
 - Javascript CLI
 - Django application

Sending metrics to Graphite

- Connect to Carbon socket (tcp/2003)
- Send your data

```
my $sock = IO::Socket::INET->new("127.0.0.1:2003");  
$sock->send("endpoint.app.metric $value $epoch\n");
```

- ...
- Profit!

Graphite Demonstration

[URL redacted]

Trending and Profiling

- What should I trend?
 - Application profiling data
 - Operational profiling data
 - Regression testing (releases)

Trending and Profiling

- Why should I trend?
 - Trends can tell you when something is *about* to break
 - ... instead of hearing from your customers that it's broken
 - Data can tell you when something *already* broke but you don't know it yet (regression)

Trending and Profiling

- Lock-step with Business/Transactional monitoring
 - Just because a host or service responds, how do you know it's working?
 - If you don't know "good", how will you recognize "bad"?
 - You don't know what might break, so collect everything now

Customer Quote

"I don't care if my servers are on fire as long as they're making me money"

Next Steps

*"Premature optimization is the
root of all evil"*

Next Steps

*"Those who ignore history are
doomed to repeat it"*

Adding Value

- Interchange/PostgreSQL profiling
 - How fast is our:
 - function `foo()` for each iteration
 - database query
 - 3rd party API service (e.g. payment gateway, social media)

Adding Value

- Interchange/PostgreSQL profiling
- How many times do we:
 - call function `foo()`
 - register a new user
 - chargeback a sale
- ... on Monday of last week? Last month?

Adding Value

- Design/User-Experience
 - A/B Testing
 - Waterfall metrics
 - Can we gather real-time waterfall metrics and submit them via AJAX?

You can't be serious!

This sounds like a lot of work.

Ok, let's make it even easier.

StatsD

- "Measure Anything, Measure Everything"
- Created and released by Etsy
- Aggregate counters and timers
- Pipeline to Graphite
- Fire-and-forget (UDP)
- <https://github.com/etsy/statsd>

StatsD

- Perl client
- <https://github.com/sivy/statsd-client>

```
use Net::StatsD::Client;  
my $c = Net::StatsD::Client->new();  
$c->increment('endpoint.customer.app.metric'); # counter  
$c->timing('endpoint.customer.app.foo', 200); # timing function, 200ms
```

- Too much activity? Sample it!

```
# sample 10%, StatsD will multiply it up  
$client->increment('endpoint.customer.app.metric', 0.1)
```

Questions?

Thank you