

# Linux Cluster in Theorie und Praxis

## Monitoring

30. November 2009

INF 1046  
Nöthnitzer Straße 46  
01187 Dresden  
0351 - 463 38783

Andy Georgi

## Foliensatz

### Verfügbarkeit der Folien

#### Vorlesungswebseite:

[http://tu-dresden.de/die\\_tu\\_dresden/zentrale\\_einrichtungen/zih/lehre/ws0910/lctp](http://tu-dresden.de/die_tu_dresden/zentrale_einrichtungen/zih/lehre/ws0910/lctp)

## Inhalt

- 1 Monitoring
- 2 Ganga  
  - Überblick
  - Architektur
  - Quellen
- 3 Nagios  
  - Überblick
  - Architektur
  - Screenshots
  - Quellen

## Monitoring

### Definition

**Monitoring** bezeichnet die systematische Erfassung oder Beobachtung von Vorgängen oder Prozessen mittels technischer Hilfsmittel. Das Ziel dieser Überwachung ist der steuernde Eingriff insofern definierte Schwellwerte über- bzw. unterschritten werden.

Auf Mehrprozessorsysteme bezogen lassen sich dementsprechend folgende Mindestanforderungen an ein Monitoring-Tool stellen:

- Zentralisierte Messwernerfassung
- Skalierbarkeit
- Minimaler Overhead
- Definition von Grenzwerten und Benachrichtigung sobald diese unter- bzw. überschritten werden

- 1 Monitoring
- 2 Ganglia
  - Überblick
  - Architektur
  - Quellen
- 3 Nagios
  - Überblick
  - Architektur
  - Screenshots
  - Quellen

- Veröffentlichung von Version 1.0.0 erfolgte 2000 durch Matthew L. Massie
- Weiterentwicklung in Zusammenarbeit mit der Universität Berkeley (Matthew L. Massie, David E. Culler) und dem San Diego Supercomputer Center (SDSC)
- Aktuelle Version: 3.1.2
- Designkriterien:
  - Skalierbarkeit
  - Fehlertoleranz
  - Erweiterbarkeit
  - Portierbarkeit
  - Minimaler Overhead
  - Wartungsaufwand unabhängig von der Anzahl der Knoten

## Architektur I

Das Ganglia-Monitoring-System setzt sich aus drei Komponenten zusammen:

- 1 Clientapplikation gmond (Ganglia Monitoring Daemon)
- 2 Serverapplikation gmetad (Ganglia Meta Daemon)
- 3 PHP-basiertes Webinterface

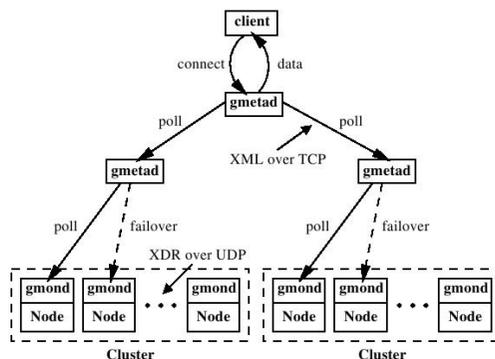


Abbildung: Ganglia Architekturüberblick [GANG04]

## Architektur II

### Ganglia Monitoring Daemon

- Installation auf jedem zu überwachenden Knoten
- Erfüllt vier Funktionen:
  - Aufnahme lokaler Messwerte
  - Verteilung der lokalen Messwerte an andere Clients
  - Entgegennahme von Messwerten anderer Clients
  - Senden des gesamten Bestands an Messwerten auf Anfrage an einen Server
- Übertragung der Statusinformationen mittels Multi- oder Unicast
- Repräsentation der Daten durch XDR
- Implementiert eine Vielzahl von Prozeduren zum Aufnehmen von Messwerten (siehe gmond -m)
- Schnittstelle für das Einbinden eigener Metriken wird bereitgestellt

## Ganglia Meta Daemon

- Installation auf einem oder wenigen Servern
- Periodische Abfrage einer oder mehrerer Monitoring Daemons oder untergeordneter Meta Daemons
- Aggregation und Speicherung der Statusinformationen in einer Datenbank
- Aufgrund der hierarchischen Struktur können mehrere Cluster von einer zentralen Stelle überwacht werden

## Quellen

-  [GANG03] University of California, Berkeley  
Ganglia Installation and Configuration, 2003  
<http://sourceforge.net/apps/trac/ganglia/wiki/Ganglia%203.1.x%20Installation%20and%20Configuration>
-  [GANG04] M. L. Massie, B.N. Chun, D. E. Culler  
The ganglia distributed monitoring system: design, implementation, and experience, Apr 2004  
<http://ganglia.info/papers/science.pdf>

## Webinterface

- Abfrage der Daten von einem Meta Daemon bei jedem Aufruf der Seite
- Knoten mit kritischen Werten werden hervorgehoben und an den Anfang der Graphenliste gesetzt

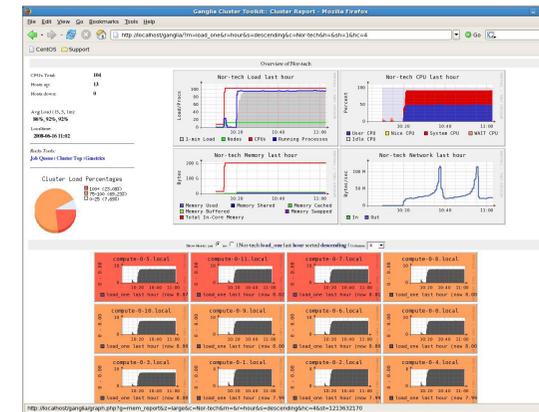


Abbildung: Screenshot Ganglia Webinterface

## Inhalt

- 1 Monitoring
  - Überblick
  - Architektur
  - Quellen
- 2 Ganglia
  - Überblick
  - Architektur
  - Screenshot
  - Quellen
- 3 Nagios
  - Überblick
  - Architektur
  - Screenshot
  - Quellen

# Überblick

- Entwicklungsbeginn März 1999 unter dem Namen NetSaint
- Autor: Ethan Galstad
- Umbenennung in Nagios im Jahr 2002 aufgrund einer Namensähnlichkeit mit einer eingetragenen Marke
- Designziele:
  - Modularer Aufbau
  - Eskalationsmanagement
  - Portier- und Skalierbarkeit
  - Selbstständige Problemlösung
  - Hochverfügbarkeit/Redundanz
  - Open Source (GPLv2)

# Architektur I

Nagios besteht im Wesentlichen aus vier Komponenten:

- 1 Nagios-Plugins
- 2 Clientapplikation (NRPE/NSCA)
- 3 Serverapplikation (Nagios)
- 4 CGI-basiertes Webinterface

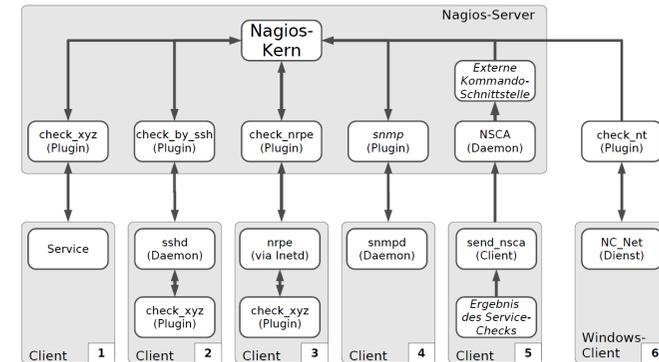


Abbildung: Nagios Architekturüberblick

# Architektur II

## Nagios-Plugins

- Verantwortlich für die Bereitstellung der Statusinformationen
- Werden entweder auf dem Server (wie z.B. `check_ping`) oder auf dem Knoten selbst (wie bspw. `check_load`) ausgeführt
- Das Plugin Paket von Nagios stellt grundlegende Funktionalitäten bereit
- Erweiterungen in Form von Binaries oder Skripten (Bash, Perl, Python... etc.) mit folgenden Eigenschaften möglich:
  - Rückgabewert darf max. eine Zeile umfassen
  - Endet mit einem der folgenden Return Codes:
    - 0 = OK
    - 1 = WARNING
    - 2 = CRITICAL
    - 3 = UNKNOWN
- Return Code abhängig von definierten Schwellwerten (Angabe mit `-w` für den Warnschwellwert und `-c` für den kritischen Schwellwert)

# Architektur III

## Clientapplikation

- Nagios stellt zwei Applikationen für die zu überwachenden Knoten bereit
- Die Wahl entscheidet gleichzeitig über die Art und Weise der Überwachung (aktiv oder passiv)
- **Nagios Remote Plugin Executor (NRPE):**
  - Ausführung der Instruktionen vom Server und Rückgabe der Ausgaben
  - Initiierung erfolgt durch den Server, daher spricht man hier von *active checks* oder *active monitoring*
- **Nagios Service Check Agent (NSCA):**
  - Initiierung der Überwachung durch den Client und selbstständiges Senden der erfassten Daten an den Server
  - Aufgrund der passiven Rolle des Servers wird diese Methode auch als *passive checks* oder *passive monitoring* bezeichnet

## Serverapplikation

- Die Serverapplikation wird, in Abhängigkeit der zu überwachenden Nodes, auf einem oder mehreren Knoten installiert
- Sammlung der Statusinformationen kann aktiv oder passiv erfolgen
- Zentralisierte Aggregation und Speicherung der gesammelten Informationen
- Erweiterung der Funktionalität durch Plugins möglich

## Webinterface

- Bereitstellung einer Laufzeitumgebung über CGI
- Für den Betrieb von Nagios nicht zwingend erforderlich, bietet sich aber als sinnvolle Erweiterung für die Auswertung der Monitoringdaten an
- Filter sollen bei der Lokalisierung von Fehlern helfen

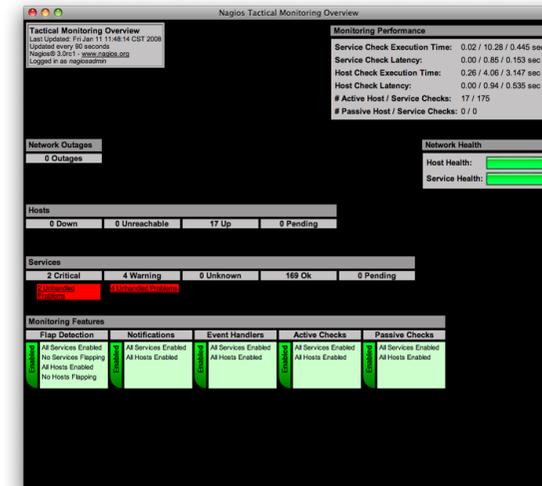


Abbildung: Nagios Tactical Monitoring Overview [NAGSCR]

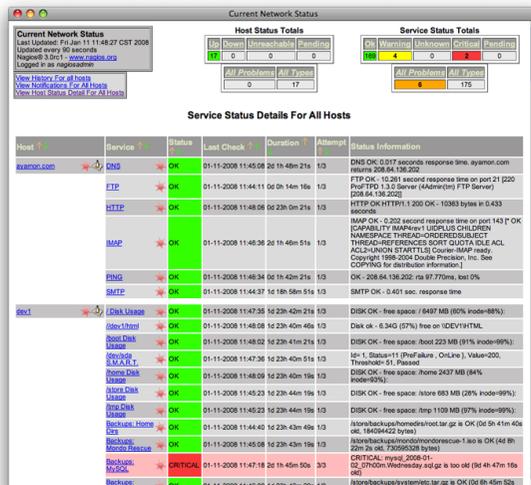


Abbildung: Service Status Details für alle Hosts [NAGSCR]

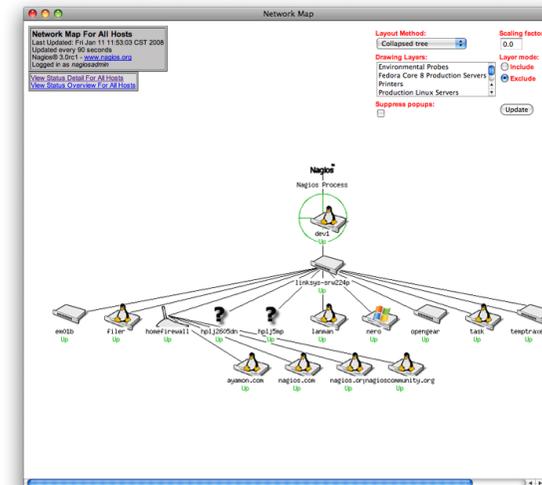


Abbildung: Tree Status Map [NAGSCR]

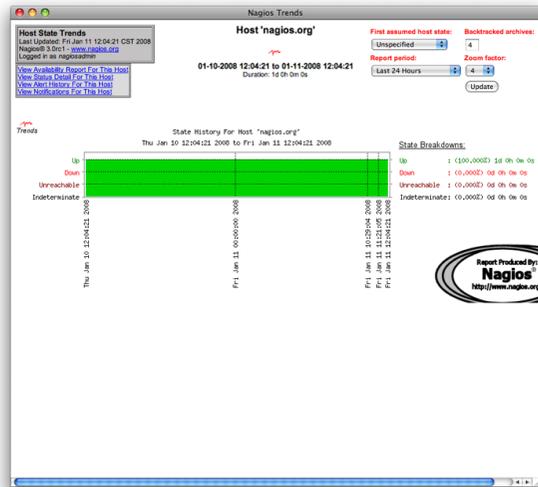


Abbildung: Host State Trends [NAGSCR]

- [NAGDOC] Nagios Core Development Team and Community Contributors  
Nagios Core Version 3.x Documentation, Jun 2009  
<http://nagios.sourceforge.net/docs/nagios-3.pdf>
- [NAGSCR] Nagios Enterprises  
Nagios Screenshots, 2009  
<http://ganglia.info/papers/science.pdf>