



System and Network Management

Mikael M. Hansen

mhansen@cs.aau.dk

2.2.57



Department of Computer Science

TOC

- Mine indlæg
- Dagens program: System and Network Management
 - Intro
 - Værktøjer
 - Eksempler fra CS

2



Department of Computer Science

Mine indlæg

- 14/2: Linux administration
- 28/2: E-Mail services og protokoller
- 4/3: Mail servers, Anti-Spam tools
- 6/3: Unix/Windows integration – Samba
- 11/3: Systemadministrations arbejdsprocesser og procedure
 - » Spørgetime (bog og praksis)
- 3/4: **Systems and Network Management**

3



Department of Computer Science

Systems and Network Management

Formål:

At introducere nogle bud på centrale værktøjer til overvågning, samt at give en praktisk vinkel på deres anvendelse og konsekvenserne heraf



Department of Computer Science

Overvågning, hvorfor ?

- Systemerne skal holdes kørende
- Er systemerne tilfredsstillende
 - Hurtige nok
 - Kan de det rigtige
 - Fleksible nok
- Hjælp til planlægning
 - Hvad skal vi sigte efter ved næste udskiftning
- Formelle krav

5



Department of Computer Science

Hvorfor?

- Ultimativt svar:
 - At give SA'erne ro i sindet.

6



Department of Computer Science

Værktøjer

- Syslog / loganalyse
- Nagios
- MRTG (RRDtool)
- Applikationerne
- Custom scripting

7

Department of Computer Science

Syslog

- Vi har snakket om den (1. kursusgang)
 - Lav en central loghost (måske endda flere)
 - Lav logrotation på den
 - F.eks. en dags logs per fil
 - Split det ud i relevante bidder
 - Brug evt. syslogs facilities som udgangspunkt eller syslog-ngs program/host parameter.
 - Brug en logparser til at ekstrahere det vigtige
 - F.eks. LogWatch (www.logwatch.org)

8

Department of Computer Science

Syslog

- Lad se på CS syslog-ng configurationen
- Sources

```
source src {
    unix-stream("/dev/log");
    file("/proc/kmsg" log_prefix("kernel: "));
    udp();
};

source local {
    unix-stream("/dev/log");
    file("/proc/kmsg" log_prefix("kernel: "));
};

source syslogng {
    internal();
};
```

9

Department of Computer Science

Syslog

- Destinations

```
destination messages { file("/var/log/messages"); };
destination secure { file("/var/log/secure"); };

destination console { user tty("root"); };

destination host_facility_info { file("/q/log/hosts/$HOST/$FACILITY-
$R_YEAR$R_MONTH$R_DAY.info"); };
destination host_facility_notice { file("/q/log/hosts/$HOST/$FACILITY-
$R_YEAR$R_MONTH$R_DAY.notice"); };

destination host_program_err { file("/q/log/hosts/$HOST/$PROGRAM-
$R_YEAR$R_MONTH$R_DAY.err"); };
```

10

Department of Computer Science

Syslog

- Filter

```
filter f_mail { facility(mail); };
filter f_debug { level(debug); };
filter f_at_least_info { level(info..emerg); };
filter vpnhosts { host("vpn1.vpn.aau.dk") or host("vpn2.vpn.aau.dk"); };
filter f_ldap { program("slapd.*"); };
filter f_squid_access { program("squid") and facility(local6); };
filter I21 { match("(MIBA-1|MIBA-2|MIBA3|MIBA4|HST)") and filter(vpnhosts)
and filter(f_localall); };
```

11

Department of Computer Science

Syslog

- Log fules

```
log { source(local); filter(f_mail); destination(maillog); };
log { source(src); filter(f_jabberd); destination(host_program_all); flags(final); };
```

12

Department of Computer Science

Logwatch

- Et script der analyser logfiler
 - Benytter et antal filtre til analysen
 - Den er pluggable mht. filtre
 - => Vi kan lave vores egne ☺
 - Perl, open source
 - => vi er sikret en vis kontinuitet
 - Laver en samlet report
 - Lad os se et eksempel

13



Department of Computer Science

Syslog

- Forudsætninger
 - At man faktisk kigger på dem
 - Handler hvis det er nødvendigt
 - Kvaliteten af loganalysen er god
 - Den må ikke miste noget
- Syslog er god til de formelle krav. Bare gem logfilerne i x antal måneder.

14



Department of Computer Science

Nagios

- "Nagios is an Open Source host, service and network monitoring program. "
(www.nagios.org)
- Opbygning
 - En Nagios host er:
 - En scheduling engine
 - Definition af hosts og services som skal overvåges
 - Et antal checks (scripts/programmer)
 - Notifications/Alerts

15



Department of Computer Science

Nagios

- Scheduling engine
 - Planlægger checks ud fra følgende kriterier
 - Konfigurationsparametre: 4 globale og 3 lokale (per service)
 - Antallet af service checks
 - Antallet af parallelle checks
 - Service state
 - Den forsøger at minimere load på både den lokale maskine og klienterne
 - Distribuerer check over hele check intervallet
 - Distribuerer check på samme host over hele intervallet

16

Department of Computer Science

Nagios

- Hosts
 - En host (server, switch ..)
 - Med IP, DNS navn
 - Hvem skal kontaktes
 - Evt. parent (hvem den er afhængig af)
 - En gruppering som den er med i (f.eks webservers)
- Service
 - Definition af et check på en host
 - Kommandoen (pluginet)
 - Hvilken host
 - Hvem skal kontaktes
 - Evt. en gruppering den er med i (SSH servers)

17

Department of Computer Science

Nagios

- Eksempler på definition af en host og en service:

```
## Host check
define host {
    use Application servers
    host_name homer.appl.cs.aau.dk
    alias homer.appl.cs.aau.dk
    address 130.225.195.4
    parents s01.a.cs.aau.dk
    check_command check-host-alive
    hostgroups Application
    contact_groups junta
}

## Service Check
define service {
    use ssh
    service_description SSH Check
    check_command check_ssh
    host_name homer.appl.cs.aau.dk
    servicegroups ssh
    contact_groups junta
}
```

Department of Computer Science

Nagios

- Checks (plugins)
 - Små programmer som ved hvordan en given service skal checkes
 - Tager et antal optioner som parameter
 - Returnerer en status for servicen
 - Ok, Warning, Critical, Unknown
 - Returner evt. performance data
 - Det ekstra som et plugin returnerer
 - Vi ser et eksempel senere
 - Man kan skrive sine egne

19



Department of Computer Science

Nagios

- Notifications
 - En advarsel sendt til den contact som er defineret på servicen
 - Det kan konfigureres hvornår den sendes
 - Når der opstår et problem
 - Når et problem bliver løst
 -
 - Hvordan den sendes kan også konfigureres
 - Mail, SMS, IM, Winpopup, Audio, --
 - » You can write your own

20



Department of Computer Science

Nagios

- Notifications, fortsat
 - De sendes kun hvis det er nødvendigt
 - F.eks. Hvis nogen allerede har acknowledged en fejl, så er der ingen grund til at advare igen.
 - Er der scheduleret down-time på servicen advarer vi heller ikke
 - En rimelig kompleks algoritme

21



Department of Computer Science

Nagios

- Add-ons
 - NagiosGrapher
 - Få tegnet grafer ud fra performance data fra checks
 - Det kigger vi på lige om lidt
 - RSS4Nagios
 - RSS feed til service checks
 - Nagios Reporter
 - Sender en daglig rapport med overblik
 - Se eksempel

22

Department of Computer Science

Nagios

- Hvad er det?
 - Et stykke software som
 - På fornuftigvis kan kører et antal check for at vedligeholde en database over tilstande på det der overvåges.
 - Ud fra ændringer i tilstande kan den advare de der ønskes, som der ønskes.
 - Kan vise status og tager imod kommandoer via en GUI.

23

Department of Computer Science

Nagios

- Lad os se på [dyret..](#)

24

Department of Computer Science

MRTG

- Multi Router Traffic Grapher
 - " You have a router, you want to know what it does all day long? Then MRTG is for you. It will monitor SNMP network devices and draw [pretty pictures](#) showing how much traffic has passed through each interface. "
 - Perl script der via SNMP henter data og smider dem i sin database
 - Plain text files (gl. måde)
 - RRD files (ny måde)
 - Kan monitorere alt som kan hentes via SNMP

25



Department of Computer Science

MRTG

- Består af
 - bin/mrtg
 - Samler data og gemmer i DB'en
 - bin/cfgmaker
 - Opretter konfigurationsfil ved at spørge den host som skal overvåges
 - bin/indexmaker
 - Genererer index.html filer til at give overblik over de grafer som genereres
 - Der er flere ..

26



Department of Computer Science

MRTG

- Plain text DB files
 - 2 dele
 - 1. line – data fra sidste kørsel
 - Resten – gennemsnit og max data fra all kørsler op til dette punkt
 - Bruges ikke mere / anbefales ikke mere

27



Department of Computer Science

MRTG

- RRD filer som backend
 - Baseret på RRDtool
 - "Round Robin Database"-tool
 - Fordele ved rrd-filer er:
 - Automatisk interpolering af værdierne
 - Så vi ikke er afhængige af at hente værdierne på et bestemt tidspunkt
 - Konsolidering
 - Kan defineres ved oprettelsen
 - Vedligeholdes automatisk
 - Round-Robin Archives
 - De bliver aldrig større

28

Department of Computer Science

MRTG

- RRD oprettelse

```
rrdtool create temperature.rrd --step 300 \  
DS:temp:GAUGE:600:-273.5000 \  
RRA:AVERAGE:0.5:1:1200 \  
RRA:MIN:0.5:12:2400 \  
RRA:MAX:0.5:12:2400 \  
RRA:AVERAGE:0.5:12:2400
```

- RRA:funktion:accuknfactor:steps:rows

29

Department of Computer Science

MRTG

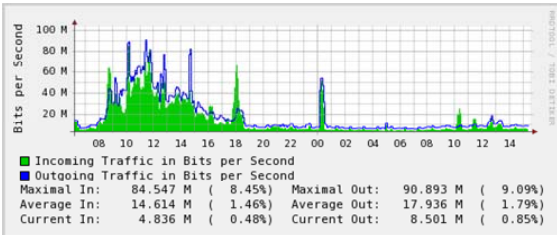
- MRTG/RRDtool har også funktion til at
 - Tegne grafer
 - Dumpe DB'en

30

Department of Computer Science

MRTG

Et eksempel:



Incoming Traffic in Bits per Second		Outgoing Traffic in Bits per Second	
Maximal In:	84.547 M (8.45%)	Maximal Out:	90.893 M (9.09%)
Average In:	14.614 M (1.46%)	Average Out:	17.936 M (1.79%)
Current In:	4.836 M (0.48%)	Current Out:	8.501 M (0.85%)

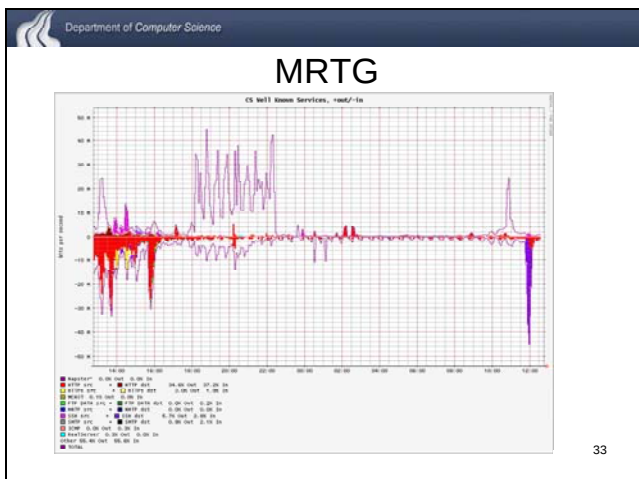
31

Department of Computer Science

MRTG

- Mere komplekst eksempel
 - Ikke MRTG men stadig RRD baseret
 - Giver en ide om mulighederne

32



Department of Computer Science

MRTG

- MRTG på CS/AAU
 - Lad os se på dyret

34



Department of Computer Science

Applikationer

- Nogle applikationer har selv
 - Statistik og performance data
 - De er ofte anvendelige når det er den pågældende applikation som er et problem
- F.eks.
 - /server-status fra Apache
 - Cachemgr.cgi fra Squid (proxy server)
 - Enterprise manager fra Oracle
 - VI klienten fra VMware

35



Department of Computer Science

Hvad gør CS

- Nagios til overvågning, alarmering
- MRTG til grafer for mest muligt
- Central loghost
 - Delvis loganalyse
- Bruger applikationerne (med vedhæng)
- Skriver selv scripts til resten
- Mangler noget... ☹

36



Department of Computer Science

Opsummering

- Få et centralt overvågningssystem
 - Som sender alarmer
 - Status reports (til arkivering)
- Lav kontinuerligt målinger
 - Helst med visualisering
 - Kapacitetsplanlægning
 - Afdækning af misbrug
- Lav central opsamling / arkivering
 - Formelle krav
 - Fremtidig reference

37

Department of Computer Science

Opsummering

- Lær de enkelte applikationer at kende
 - Brug deres egne værktøjer
- Skriv selv scripts til resten
- Når noget nyt opstår så tilføj det.
- Lav en værktøjssamling/Webside
 - Til overblikket
 - Specielt hvis der er flere SA'er
- Husk: et nyt produkt/service er ikke i produktion før det er overvåget!

38

Department of Computer Science

Spørgsmål?

Department of Computer Science

Tak for denne gang ☺
