Il sistema di monitoring GARR: presente e futuro.

Un viaggio alle radici di
GARR Integrated Networking Suite

giovanni.cesaroni@garr.it

WORKSHOP GARR_08

GARR-X: il futuro della Rete _ Milano 1-4 aprile 2008

Contenuti

- GINS summary
  - Gli scopi
  - Stato dell'arte
  - Strumenti SW utilizzati
- Viaggio nella visualizzazione
  - Statistiche
  - Nuova weathermap del backbone

Contenuti

- Viaggio nell'acquisizione, SNMP applicato
  - RFC
  - Eseguire una query
  - Un rudimentale ma efficace BGP monitor in 5 slide
  - Monitorare i costi OSPF
  - MPLS LSP monitor

Contenuti

- Viaggio nello storage, RRD applicato
  - come e' fatto un file RRD
  - come generare statistiche *NON COMPRESSE*
- Futuro

GINS scopi

1- Controllare il grado di servizio ad ogni livello dell'infrastrutura

2- Integrare i servizi e svilupparne di nuovi

3- Permettere un accesso facile all'informazione

GINS stato dell'arte

Elementi costituenti:

GINS stato dell'arte

~ 200 apparati (router ,switch, wdm)

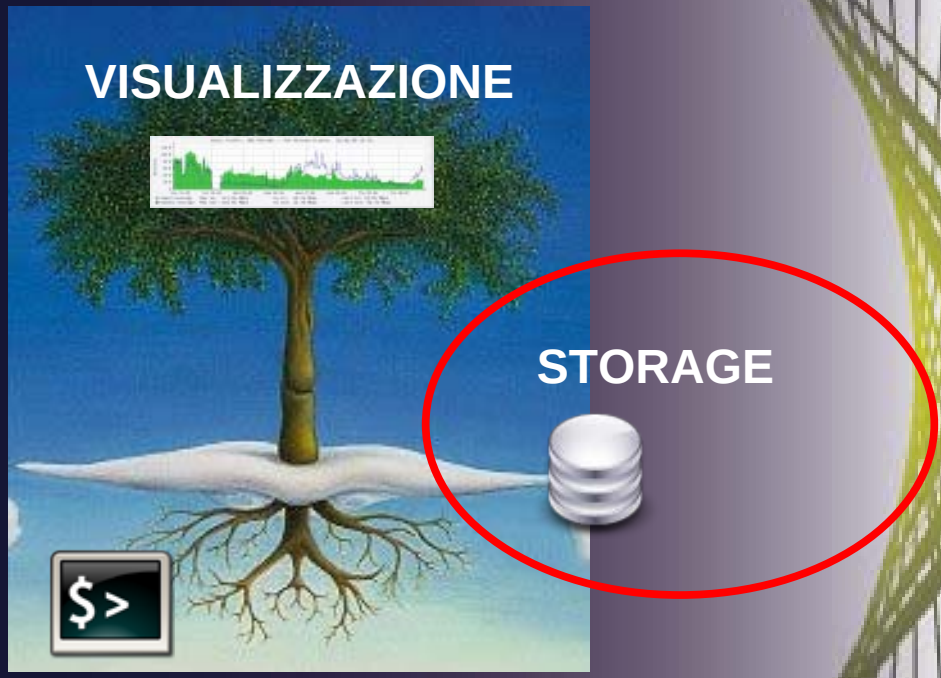
~ 1600 circuiti

- Multicast
- BGP
- OSPF
- IPv4, IPv6
- MPLS
- Sonet/SDH
- Lambda

GINS stato dell'arte

5335 file RRD

automatizzazione



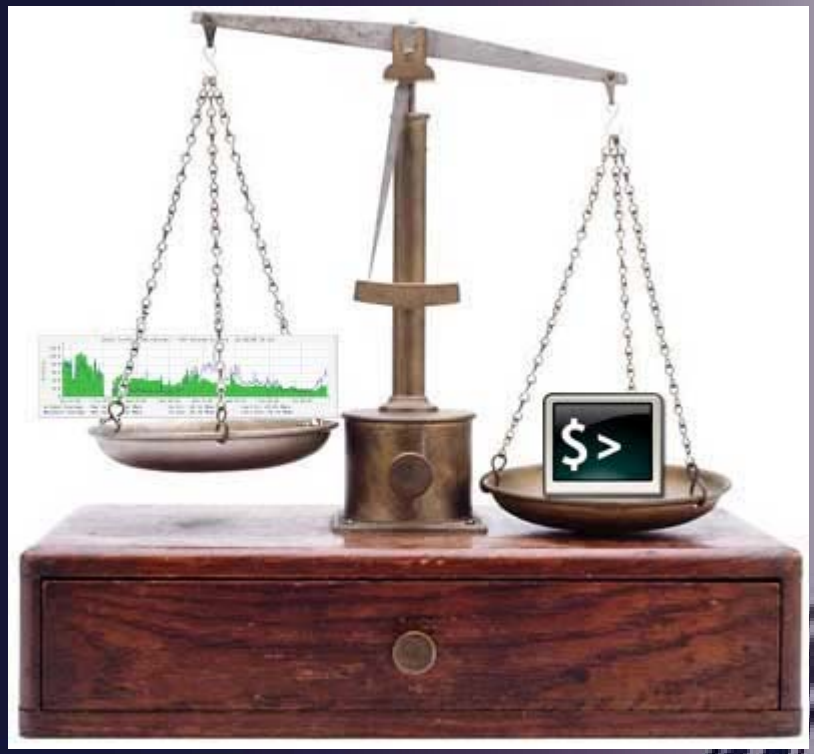
GINS stato dell'arte

ACQUISIZIONE, ELABORAZIONE

Premessa

Nella presentazione non manchera' il codice

= Strumenti pronti per l'uso



Strumenti SW utilizzati

Scheduler:
Cron

Strumenti di visualizzazione:
PHP, HTML, Javascript, Ajax, SVG

Strrumenti di storage:
MySQL, File, RRD

Strumenti di gestione ed elaborazione dati:
AWK, Bash, PHP

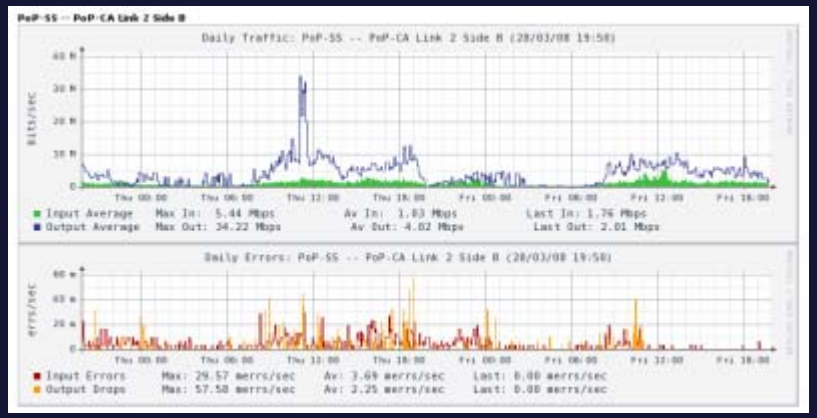
Strumenti di acquisizione dati:
MRTG, SNMP polls, ping

Viaggio nella visualizzazione

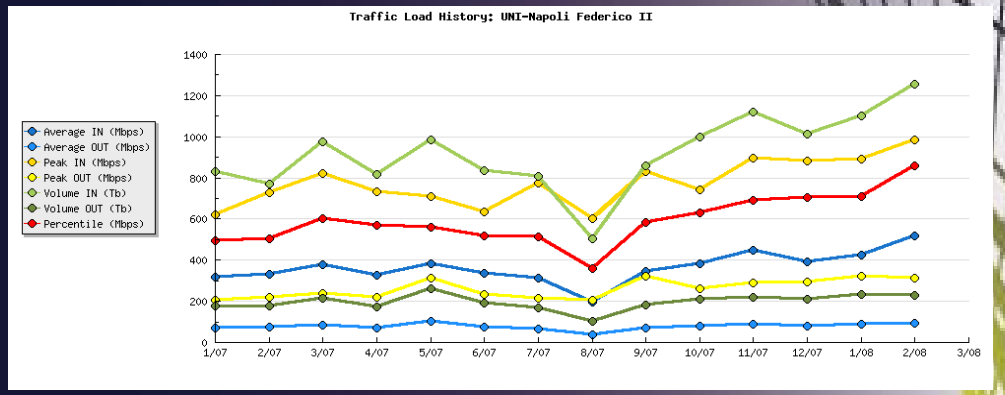
ACQUISIZIONE, ELABORAZIONE

Statistiche:

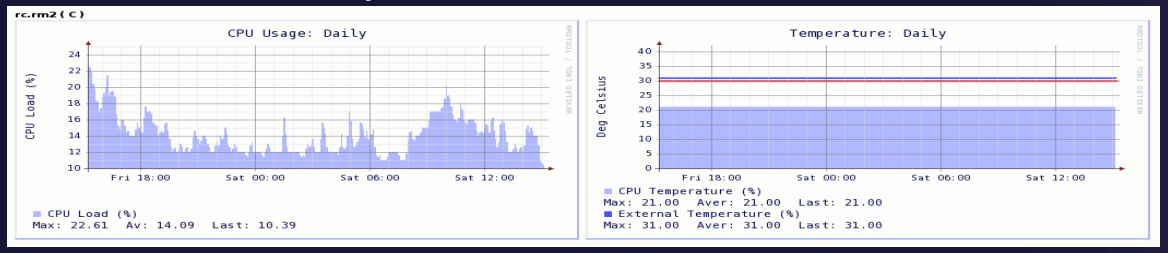
Traffic, Input errors & output drops



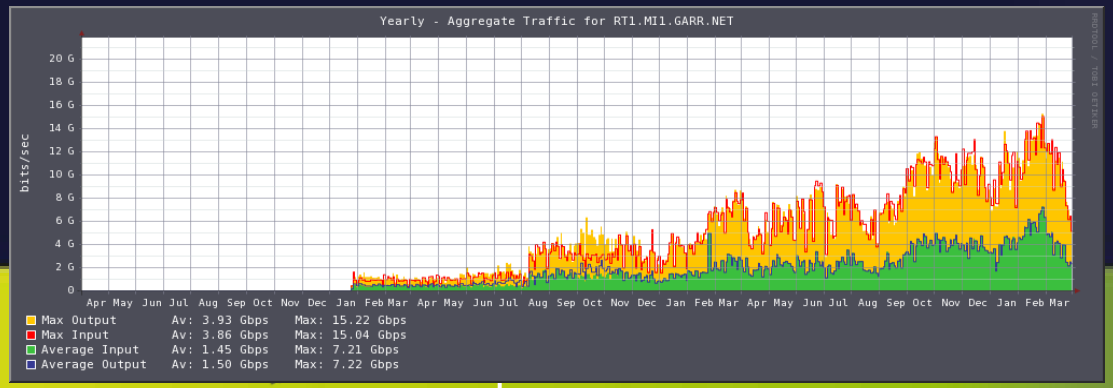
Uncompressed traffic



CPU load & temperature

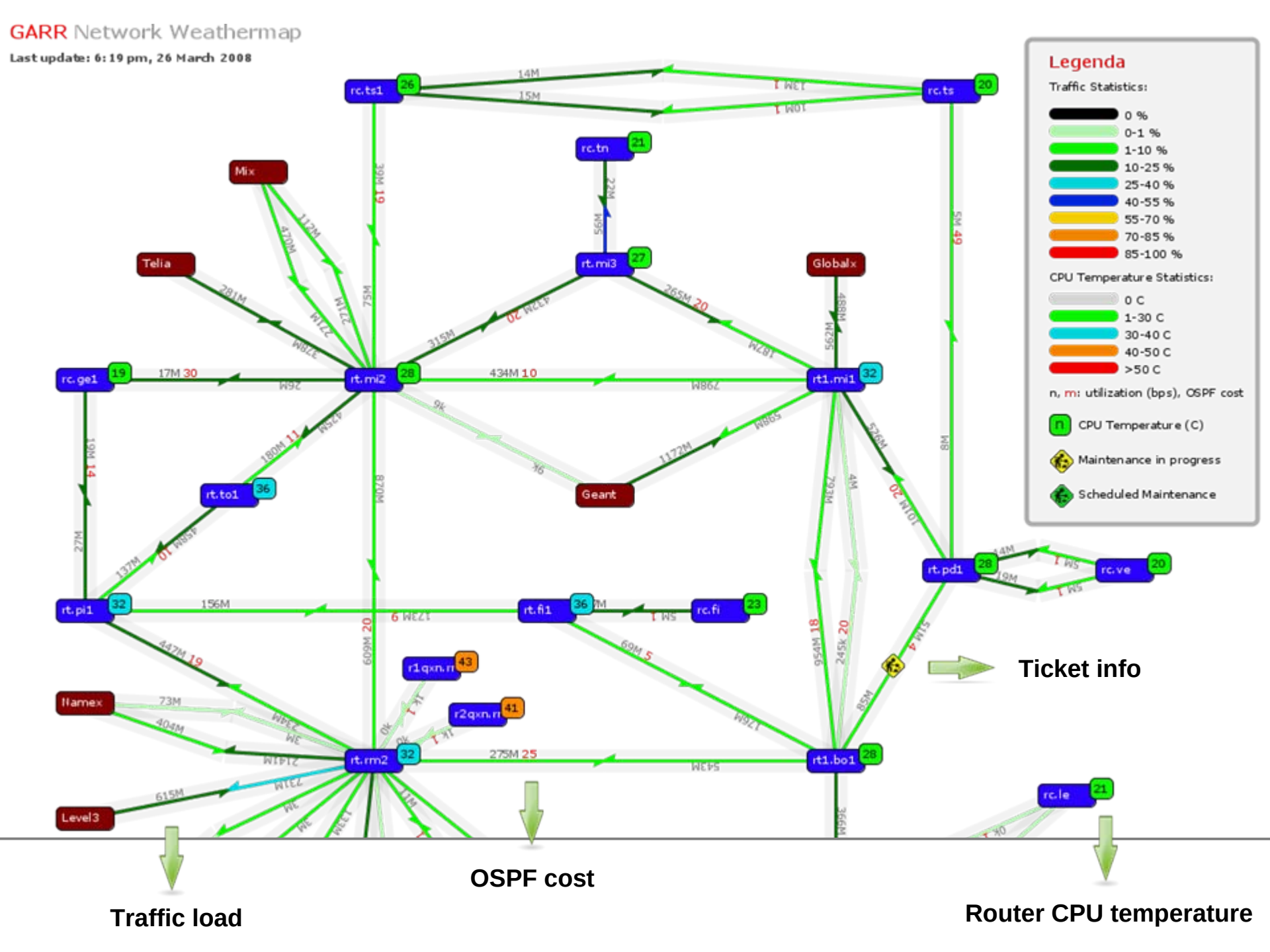


Router/Users aggregate traffic & peaks



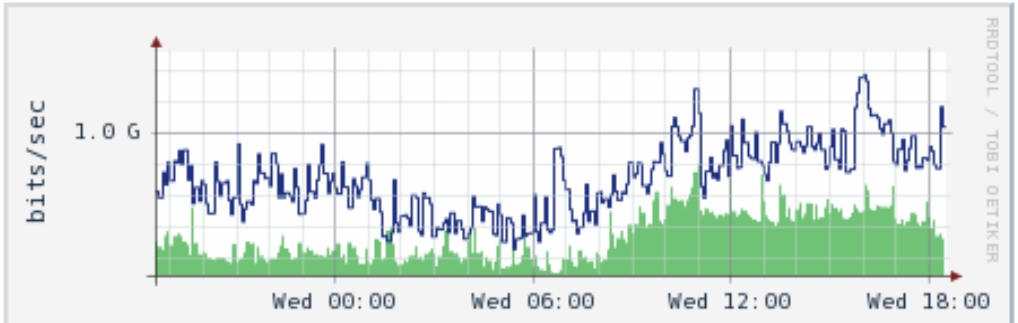
La nuova weathermap del backbone:

- accesso visuale allo stato della rete nel suo complesso
- flessibilita'
- leggerezza
- bellezza


GARR Network Weathermap
Last update: 6:19 pm, 26 March 2008
Legenda
Traffic Statistics:
0 %
0-1 %
1-10 %
10-25 %
25-40 %
40-55 %
55-70 %
70-85 %
85-100 %
CPU Temperature Statistics:
0 C
1-30 C
30-40 C
40-50 C
>50 C
n, m: utilization (bps), OSPF cost
CPU Temperature (C)
Maintenance in progress
Scheduled Maintenance
rc.ts1
rc.ts
rc.tn
Mix
Telia
rt.mi3
Globalx
rc.ge1
rt.mi2
rt1.mi1
rt.to1
Geant
rt.pd1
rc.ve
rt.pi1
rt.fi1
rc.fi
r1q×n.rt
r2q×n.rt
Namex
rt.rm2
rt1.bo1
Level3
rc.le
Ticket info
OSPF cost
Traffic load
Router CPU temperature

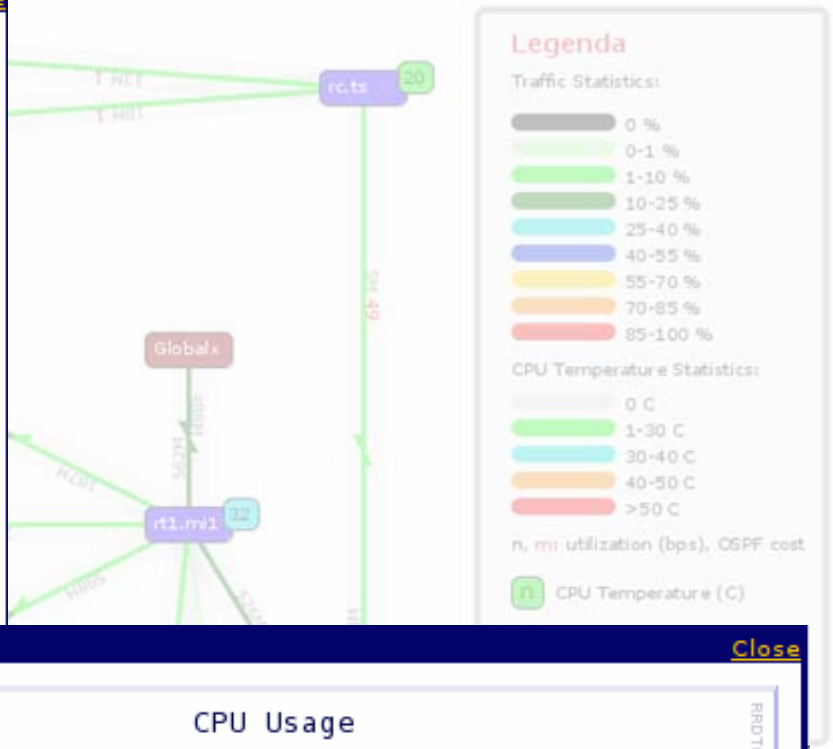
PoP-MI2 -- PoP-MI1

Close



Input	Output
Max: 792.21 Mbps	Max: 1.37 Gbps
Av: 332.59 Mbps	Av: 704.35 Mbps
Last: 315.86 Mbps	Last: 1.04 Gbps

Click here to display statistics details
Click here to display statistics details with errors
Click here to display statistics details with peaks



Ticket Info

Close

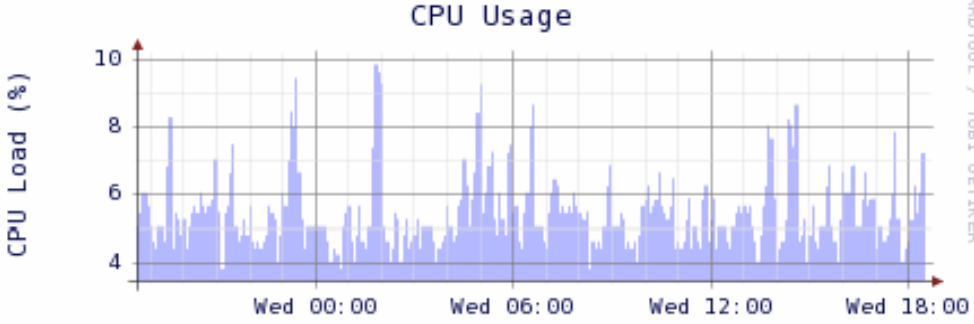
Ticket Number: 7019
Tcket Status: Update
Tcket Type: Unscheduled
Tcket Source: GARR
Problem Sort: LF
Problem Fixer: GARR-NOC
Problem Description:

Problem Start: 13-11-2007 13:59
Action to Fix:
[14/11/2007,11:06,dipeo]: In sostituzione un componente passivo del DWDM di Bologna. Riscontrata la necessita' di sostituire il pezzo nella notte, il componente stesso e' partito da Firenze alle 8.00. Si prevede il ripristino in un paio d'ore
[13/11/2007,19:48,pellegri]: sollecitato Infracom, tempi di ripristino non comunicati
[13/11/2007,18:12,pellegri]:
[13/11/2007,15:35,pellegri]: rilevato problema ad un apparato fra Occhiobello e Ferrara, uscita squadra
[13/11/2007,14:04,pellegri]: contattato Infracom, in attesa di informazioni

Display Ticket Details

rt1.mi1

Close

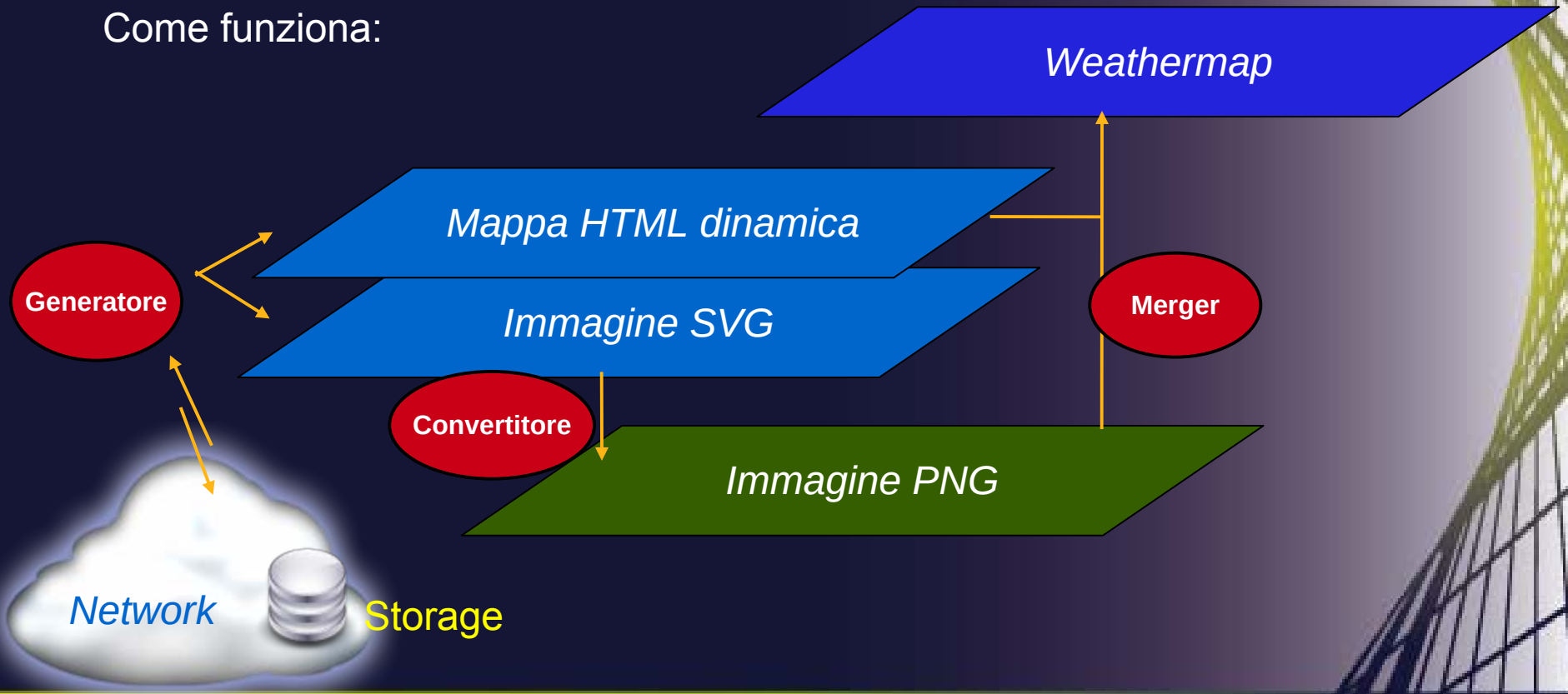


CPU Load (%)
Max: 9.82 Av: 5.42 Last: 7.18

Click here to display the CPU load and temperature statistics
Click here to display the aggregate traffic statistics

La nuova weathermap del backbone:

Come funziona:



Viaggio nel codice

SNMP, gli RFC:

n. 1441

Introduction to version 2 of the Internet-standard Network Management Framework

n. 2578

Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)

n. 1213 (updates 2011,2013,2013)

Management Information Base for Network Management of TCP/IP-based internets: MIB-II

SNMP Poll:

snmpget -v2c -c <community> <router> <Object Identifier OID>

snmpwalk -v2c -c <community> <router> <parte di un OID>

La risposta alla query:

<OID> = <tipo di dato>: <valore>

BGP Monitor 1/5:

Stato del Peer BGP: 1.3.6.1.2.1.15.3.1.2 (RFC 1269)

AS del Peer BGP: 1.3.6.1.2.1.15.3.1.9 (RFC 1269)

BGP Monitor 2/5:

Stato del Peer BGP: 1.3.6.1.2.1.15.3.1.2

```
snmpwalk -v2c -c <community> <router> 1.3.6.1.2.1.15.3.1.2 |
      awk -F 'SNMPv2-SMI::mib-2.15.3.1.2.' '{print $2}' |
      awk -F ' = INTEGER: ' '{
       if($2=="1"){status=sprintf("Idle");};
       if($2=="2"){status=sprintf("Connect");};
       if($2=="3"){status=sprintf("Active");};
       if($2=="4"){status=sprintf("Opensent");};
       if($2=="5"){status=sprintf("Openconfirm");};
       if($2=="6"){status=sprintf("Established");};
       print $1,status;}'
```

restituisce una lista di cui le righe contengono:
<IP del Peer> <stato>

BGP Monitor 3/5:

AS del Peer BGP: 1.3.6.1.2.1.15.3.1.9

```
snmpwalk -v2c -c <community> <router> 1.3.6.1.2.1.15.3.1.9 |
      awk -F 'SNMPv2-SMI::mib-2.15.3.1.9.' '{print $2}' |
      awk -F ' = INTEGER: ' '{print $1,$2;}'
```

restituisce una lista di cui le righe contengono:
<IP del Peer> <AS>

BGP Monitor 4/5:

Una rudimentale ma efficace implementazione

In /<path>/BGPmon.sh

```
#!/bin/bash
snmpwalk -v2c -c <community> <router> 1.3.6.1.2.1.15.3.1.2 |
awk -F 'SNMPv2-SMI::mib-2.15.3.1.2.' '{print $2}' |
awk -F ' = INTEGER: ' '{
        if($2!="6"){alarm=sprintf("Problema del Peer");};
        print alarm,$1;}'
```

In crontab

```
MAILTO="giovanni.cesaroni@garr.it"
0-55/5 * * * * /<path>/BGPmon.sh
```

BGP Monitor 5/5:

CISCO-BGP4-MIB
Accepted prefixes per Peer
1.3.6.1.4.1.9.9.187.1.2.4.1.1.<IP>.1.1 (.1.1 = IPv4 Unicast)
Advertised prefixes per Peer
1.3.6.1.4.1.9.9.187.1.2.4.1.6.<IP>.1.1

BGP4-V2-MIB-JUNIPER
Received prefixes per Peer
1.3.6.1.4.1.2636.5.1.1.2.6.2.1.7.<Peer Index>.1.1
Advertised prefixes per Peer
1.3.6.1.4.1.2636.5.1.1.2.6.2.1.10.<Peer Index>.1.1
Accepted prefixes per Peer
1.3.6.1.4.1.2636.5.1.1.2.6.2.1.8.<Peer Index>.1.1
Peer Index from:
1.3.6.1.4.1.2636.5.1.1.2.1.1.1.14

OSPF Cost Monitor:

Costo OSPF di un link: 1.3.6.1.2.1.14.8.1.4.<IP Address>.0.0
(RFC 1850)

```
snmpwalk -v2c -c <community> <router> 1.3.6.1.2.1.14.8.1.4 |
        grep '.0.0 =' |
        awk -F '.0.0 = INTEGER: ' '{print $1,$2}' |
        awk -F 'SNMPv2-SMI::mib-2.14.8.1.4.' '{print $2}'
```

restituisce una lista di cui le righe contengono:
<IP Address> <Costo OSPF>

MPLS LSP Monitor (su Juniper):

Per ottenere le informazioni su un LSP dobbiamo conoscere l'indice che identifica l'LSP (<LSP index>),
una creatura dalle sembianze inumane

BO1-MI1-VPN :
.66.79.49.45.77.73.49.45.86.80.78.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0

snmpget -v2c -c <comunity> <router>
<mplsLspState>.<LSP index>

1 = unknown
2 = up
3 = down

Le informazioni:

mplsLspName 1.3.6.1.4.1.2636.3.2.3.1.1
mplsLspPathChanges 1.3.6.1.4.1.2636.3.2.3.1.10
mplsLspLastPathChange 1.3.6.1.4.1.2636.3.2.3.1.11
mplsLspConfiguredPaths 1.3.6.1.4.1.2636.3.2.3.1.12
mplsLspStandbyPaths 1.3.6.1.4.1.2636.3.2.3.1.13
mplsLspOperationalPaths 1.3.6.1.4.1.2636.3.2.3.1.14
mplsLspFrom 1.3.6.1.4.1.2636.3.2.3.1.15
mplsLspTo 1.3.6.1.4.1.2636.3.2.3.1.16
mplsPathName 1.3.6.1.4.1.2636.3.2.3.1.17
mplsPathType 1.3.6.1.4.1.2636.3.2.3.1.18
mplsPathExplicitRoute 1.3.6.1.4.1.2636.3.2.3.1.19
mplsLspState 1.3.6.1.4.1.2636.3.2.3.1.2
mplsPathRecordRoute 1.3.6.1.4.1.2636.3.2.3.1.20
mplsPathBandwidth 1.3.6.1.4.1.2636.3.2.3.1.21
mplsPathCOS 1.3.6.1.4.1.2636.3.2.3.1.22
mplsPathInclude 1.3.6.1.4.1.2636.3.2.3.1.23
mplsPathExclude 1.3.6.1.4.1.2636.3.2.3.1.24
mplsPathSetupPriority 1.3.6.1.4.1.2636.3.2.3.1.25
mplsPathHoldPriority 1.3.6.1.4.1.2636.3.2.3.1.26
mplsPathProperties 1.3.6.1.4.1.2636.3.2.3.1.27
mplsLspOctets 1.3.6.1.4.1.2636.3.2.3.1.3
mplsLspPackets 1.3.6.1.4.1.2636.3.2.3.1.4
mplsLspAge 1.3.6.1.4.1.2636.3.2.3.1.5
mplsLspTimeUp 1.3.6.1.4.1.2636.3.2.3.1.6
mplsLspPrimaryTimeUp 1.3.6.1.4.1.2636.3.2.3.1.7
mplsLspTransitions 1.3.6.1.4.1.2636.3.2.3.1.8
mplsLspLastTransition 1.3.6.1.4.1.2636.3.2.3.1.9

MPLS LSP Monitor (su Juniper):

Ricavare l'index

B O 1 - M I .....

traduzione in ascii decimale

.66.79.49.45.77.73.49.45.86.....

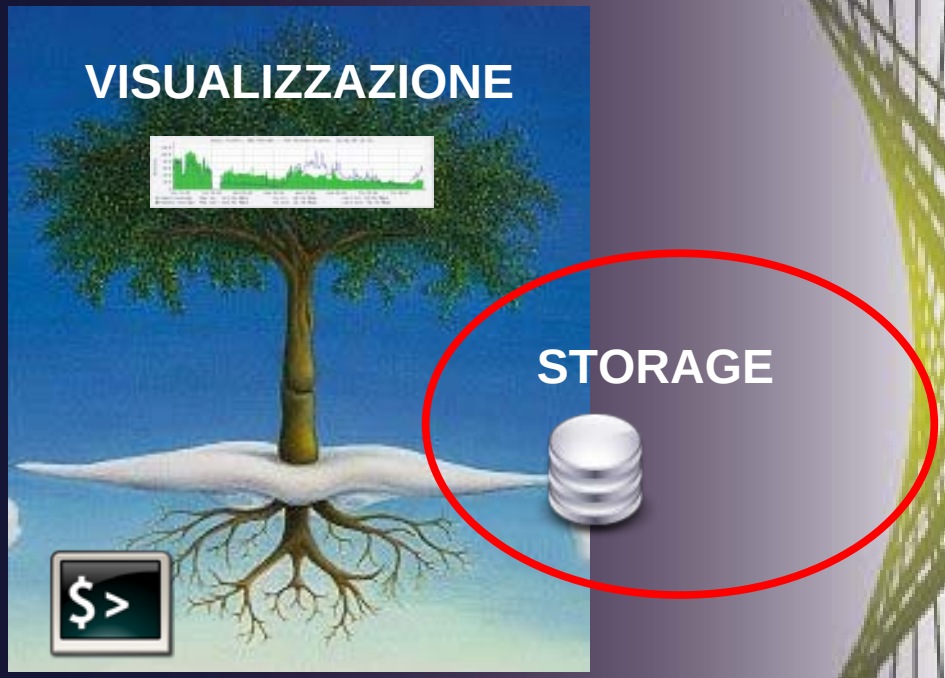
```
<?
$name=$argv[1];
$oid=name2oid($name);
print $name.": ".$oid."\n";

function name2oid($string) {
   $hex = '';
   $len = strlen($string);
   for ($i = 0; $i < $len; $i++) {
      $hex .= ".".str_pad(ord($string[$i]), 2, 0, STR_PAD_LEFT);
   }
   $npoints=32-$len;
   for ($i=0;$i<$npoints;$i++){
      $hex .= ".0";
   }
   return $hex;
}
?>
```

ognuno ha il suo convertitore:

$ php name2oid.php BO1-MI1-VPN

BO1-MI1-VPN: .66.79.49.45.77.73.49.45.86.80.78.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0

Viaggio nello storage



ACQUISIZIONE, ELABORAZIONE

RRD tool

tutte le informazioni si trovano qui:

http://oss.oetiker.ch/rrdtool/

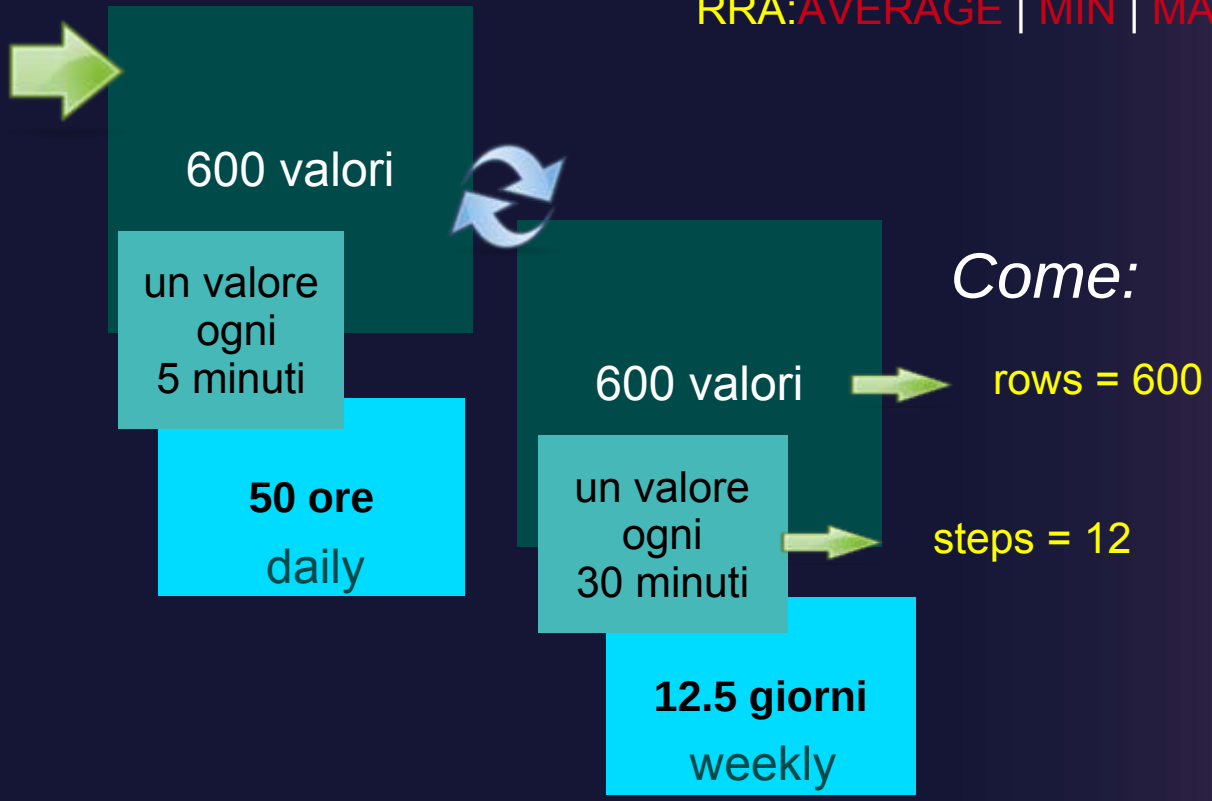
thanks to Tobias Oetiker

RRD file:

una *possibile* e *classica* struttura temporale:

RRD file:

Cosa e *come* conservare:



Cosa:

RRA:AVERAGE | MIN | MAX | LAST:xff:steps:rows

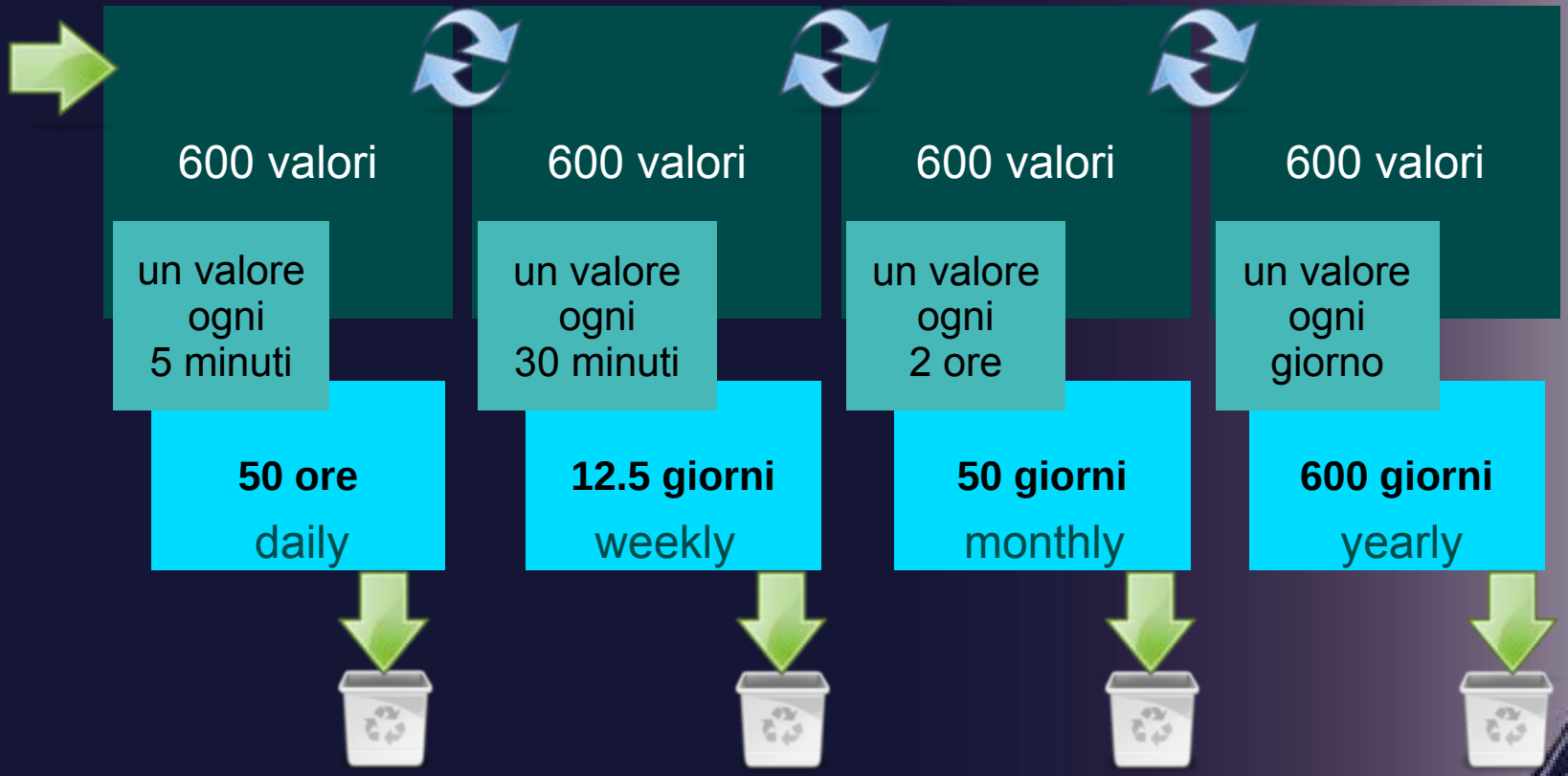
Come:

rows = 600

steps = 12

RRD file:

L'aggiornamento: per ogni valore che entra uno ne esce, in ogni blocco

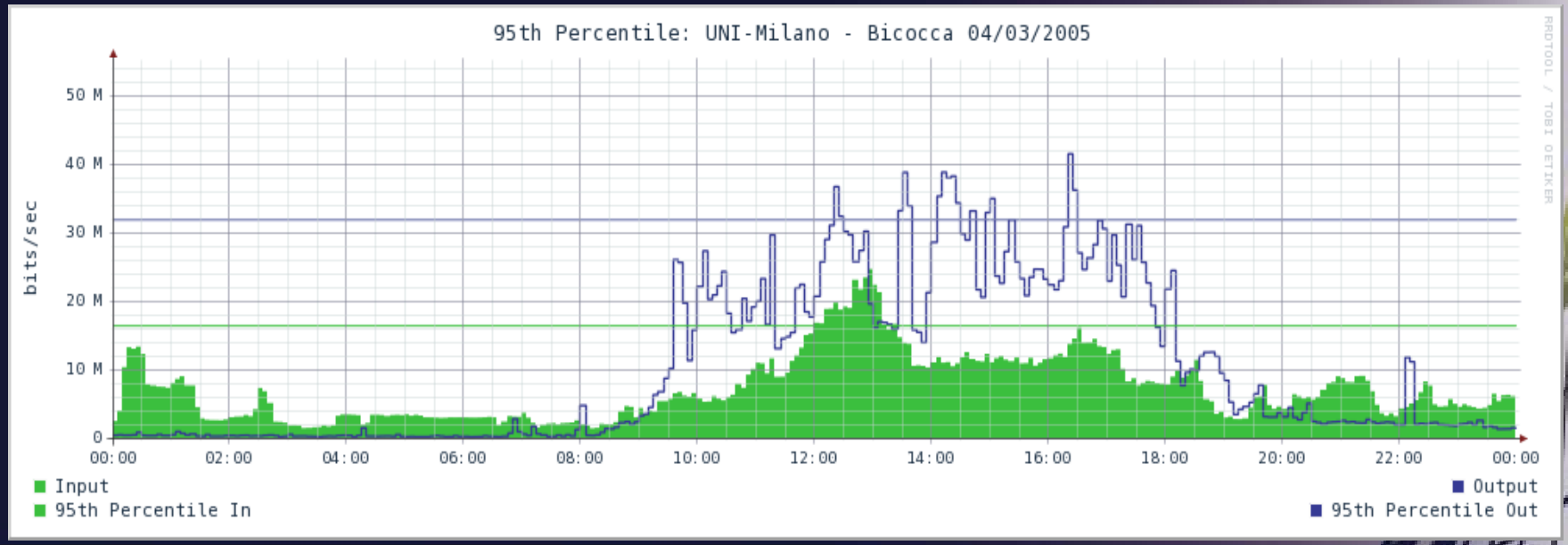


Statistiche non compresse:

Statistiche non compresse:

Daily Uncompressed Traffic Statistics:

Select Date:	04/03/2005		
Select the Site:	UNI-Milano - Bicocca		
Period:	**Start:** 0 AM	**For:** 24 Hours	
Show Reset			



Statistiche non compresse:

La creazione del file RRD non compresso (una volta l'anno):

```
rrdtool create <destination.rrd>
>       --start <qualche anno fa> --step 300
>       DS:in:GAUGE:600:U:U DS:out:GAUGE:600:U:U
>       RRA:LAST:0.5:1:105408
```

Estrazione ed inserimento dei dati (ogni ora):

```
rrdtool fetch <source.rrd> --end now-600s --start now-4200s AVERAGE |
      awk -F ' ' 'BEGIN {x=0;}{x++; if (x>2){ print $1 $2":"$3 } }' |
      xargs rrdtool update <destination.rrd>
```

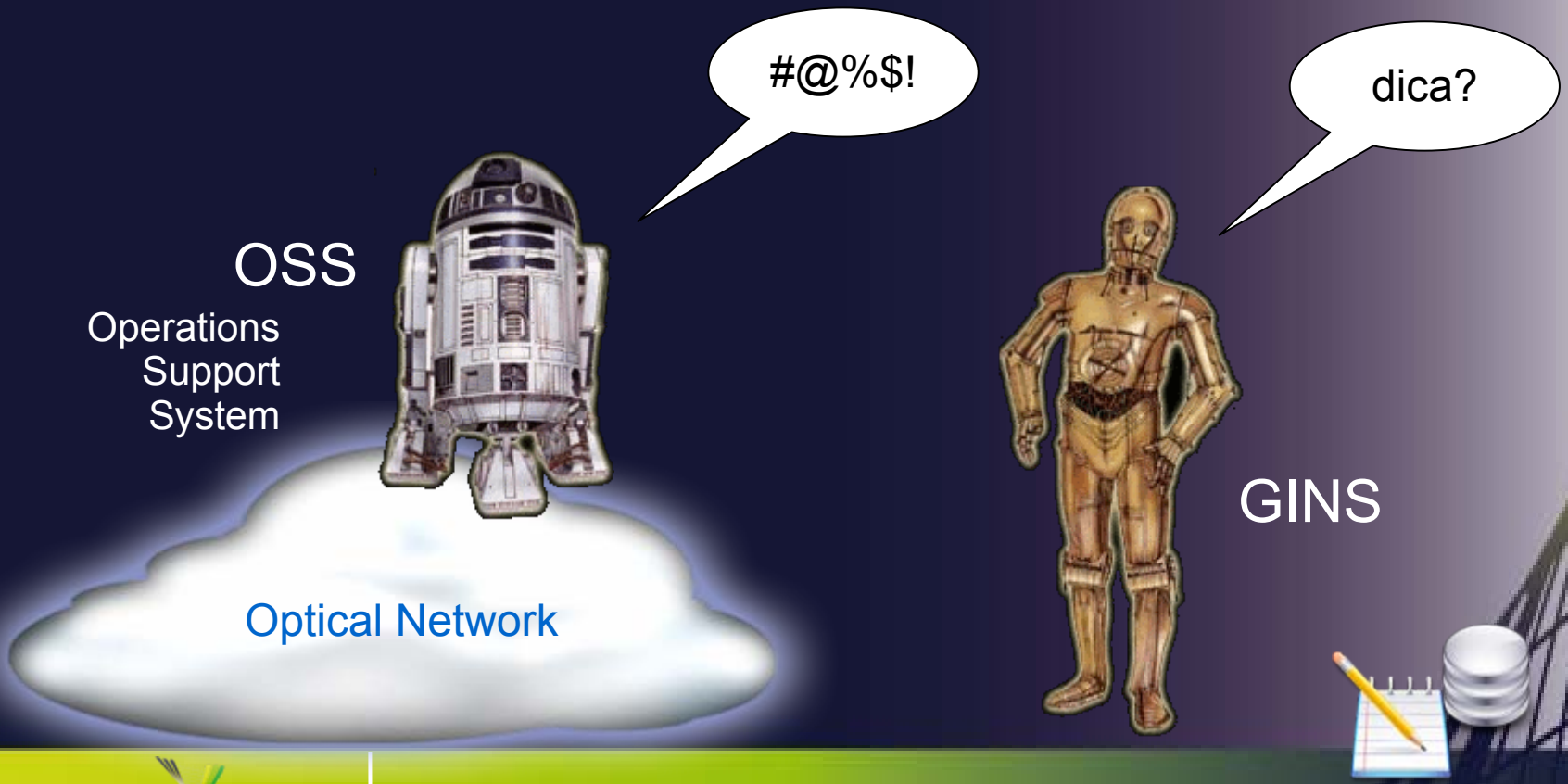
si interagisce con RRD grazie alle API di rrdtool:
create, fetch, tune, graph, dump, restore, etc.

Bolle in pentola:

- Packaging: a partire da singoli blocchi, quali weathermap, BGPmon, etc
- Analisi dei flussi (*netflow*)
- Project monitoring:
  generazione automatica di un sistema di monitoring dedicato ad uno specifico progetto
- PoP monitoring:
  tutto cio' che succede nel PoP: temperature, alimentazioni, mappe, etc

Un futuro *presente*:

• Definizione dell'architettura hw/sw per l'integrazione dei sistemi di gestione degli apparati ottici



Hanno contribuito:

sw.dev
NOC
Operations
Planning

Ringraziamento speciale a:

Rino Nucara, per il contributo nello scrivere un *Mr* codice

Domande:

Per voi:

cosa pensate di uno strumento di condivisione del sapere,
di tipo wiki / blog / forum, deedicato all'amministrazione di rete?

Per me:

?

Contatti:

sw.dev@garr.it

giovanni.cesaroni@garr.it