

Network Awareness and perfSONAR

Why we want it.

What are the challenges?

Where are we going?

Shawn McKee / University of Michigan

OSG AHG - US CMS Tier-2 Facilities Workshop

March 23, 2015

Initial Remarks on perfSONAR

- ❄ **Over the last few years WLCG sites have converged on perfSONAR as their way to measure and monitor their networks for data-intensive science.**
 - ❑ Not easy to get global consensus but we have it now. Now mandated by all LHC VOs
- ❄ **Globally distributed infrastructure requires network measurement to:**
 - ❑ Understand and baseline network capacity between resource sites
 - ❑ Identify and quickly fix network problems
 - ❑ Inform higher-level services for decision support in network use
- ❄ **A Modular dashboard (MaDDash) is critical for “visibility” into networks. We can’t manage/fix/respond-to problems if we can’t “see” them.**
- ❄ **OMD/Check_mk (used to monitor and verify the state of many globally distributed perfSONAR services) is required to maintain the overall proper functioning of the monitoring infrastructure.**
- ❄ **The development of the “mesh-configuration” and corresponding GUI interface was critical to creating a scalable, manageable deployment for WLCG/OSG**
- ❄ **Having perfSONAR fully deployed with a global dashboard is giving us powerful options for better management and use of our network**

Vision for perfSONAR-PS in WLCG/OSG

❄ Primary Goals:

- ❄ Find and isolate “network” problems; alerting in a timely way
- ❄ Characterize network use (base-lining)
- ❄ Provide a source of network metrics for higher level services
- ❄ **First step:** get monitoring in place to create a baseline of the current situation between sites
- ❄ **Next:** continuing measurements to track the network, alerting on problems as they develop
- ❄ **Choice of a standard “tool/framework”:** perfSONAR
 - ❄ We want to benefit from the R&E community consensus
- ❄ perfSONAR’s purpose is to aid in network diagnosis by allowing users to characterize and isolate problems. It provides measurements of network performance metrics over time as well as “on-demand” tests.

- ❄ Since it's part of my presentation title we should cover this! (More on this on my Wednesday talk)
- ❄ Networks underlie our distributed computing model but are historically only **indirectly** visible. This led many to feel most problems with a WAN involved were network problems (and sometimes that was true).
- ❄ perfSONAR is part of an evolving infrastructure where the network plays a much more visible role. With perfSONAR we can monitor our network, understand capacity, find bottlenecks and detect problems. It is NOT the only thing we need.
- ❄ With Software Defined Networking slowly creeping into our network hardware we will have more opportunity in the future to integrate the network we need into our end-to-end systems.
- ❄ **The goal is to make the network visible and controllable to improve our infrastructure, avoid congestion, work around failures and improve efficiency.**

Challenges Deploying/Supporting perfSONAR

- ❄ So I think we would all agree there are good reasons to deploy and use perfSONAR (pause for dissent.....)
 - ❑ But I also know some sites have had more “work” than planned in doing this
- ❄ Our original goal was to provide something that was as much “set-it-and-forget-it” as possible.
 - ❑ **Hard: A complex set of services deployed in many ways on heterogeneous hardware.**
 - ❑ Compared to other services required for LHC it is still not very difficult to deploy or upgrade.
 - ❑ Ease of installation and management still a primary motivator for us.
- ❄ **Issues encountered [solutions]**
 - ❑ Difficulty installing because of hardware specifics or need to integrate with build systems. [<https://twiki.opensciencegrid.org/bin/view/Documentation/DeployperfSONAR>]
 - ❑ Misconfiguration apparently still too easy to have happen [Mesh-config, check_mk tests]
 - ❑ Tests stop running or stop gathering metrics [New 3.4.2 version coming]
 - ❑ Disks filling, services take too many resources [Reconfig logging; default 3.4.2; mem too small]

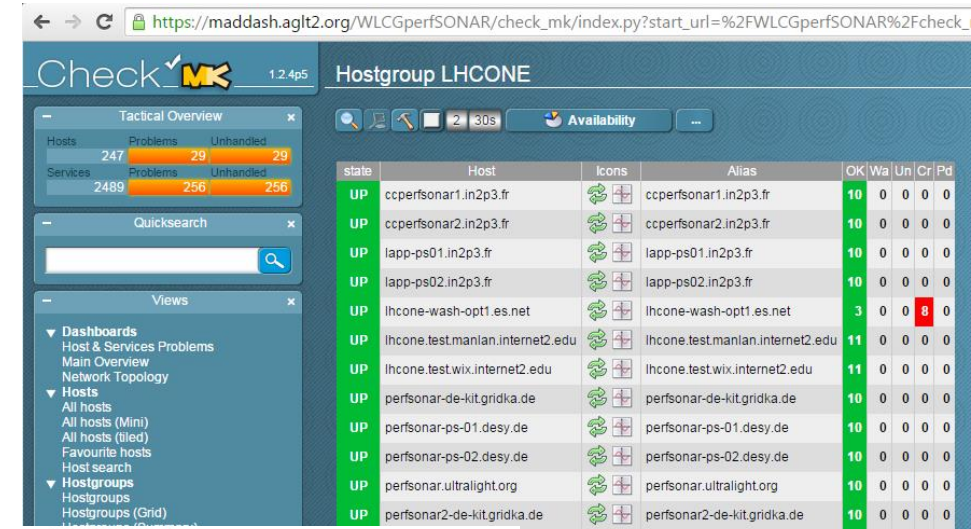
- ❄ We are hard at work identifying issues and working with the perfSONAR developers to get us where we want to be.
 - ❑ WLCG/OSG is larger than other user communities. Because of our scale and use-cases we often identify weaknesses not easily found by smaller deployments.
- ❄ We are providing “user/admin” interfaces to make the system more valuable for everyone.
 - ❑ Service/host monitoring via OMD/Check_MK (see URLs at end)
 - ❑ MaDDash to expose results
 - ❑ Central OSG Network Datastore to provide one-stop access to all metrics
- ❄ We have a testbed to try new versions of perfSONAR out at suitable scale before new releases are made.
 - ❑ Nebraska participates; very helpful to identify subtle issues; being used to vet 3.4.2RC now.
 - ❑ See MaDDash at <https://maddash.aglt2.org/maddash-webui/index.cgi?dashboard=perfSONAR%20Testbed>

❄ Our WLCG working group (Network and Transfer Metrics WG) is organizing and managing our perfSONAR deployment.

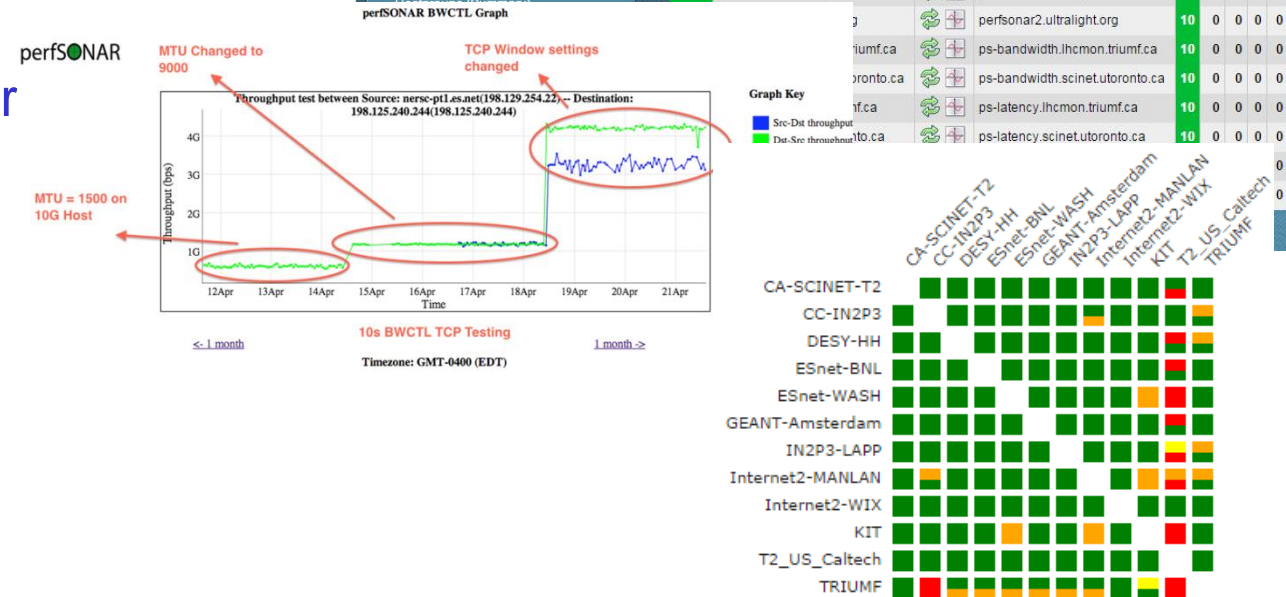
- ❑ <https://twiki.cern.ch/twiki/bin/view/LCG/NetworkTransferMetrics>
- ❑ Email [wlcg-perfsonar-support 'at' cern.ch](mailto:wlcg-perfsonar-support@cern.ch)

❄ OSG is providing our core services:

- ❑ MaDDash GUI for perfSONAR metrics
- ❑ OMD/Check_Mk (Nagios system to monitor services)
- ❑ Network datastore (using perfSONAR measurement archive)
- ❑ Mesh-creation and management system



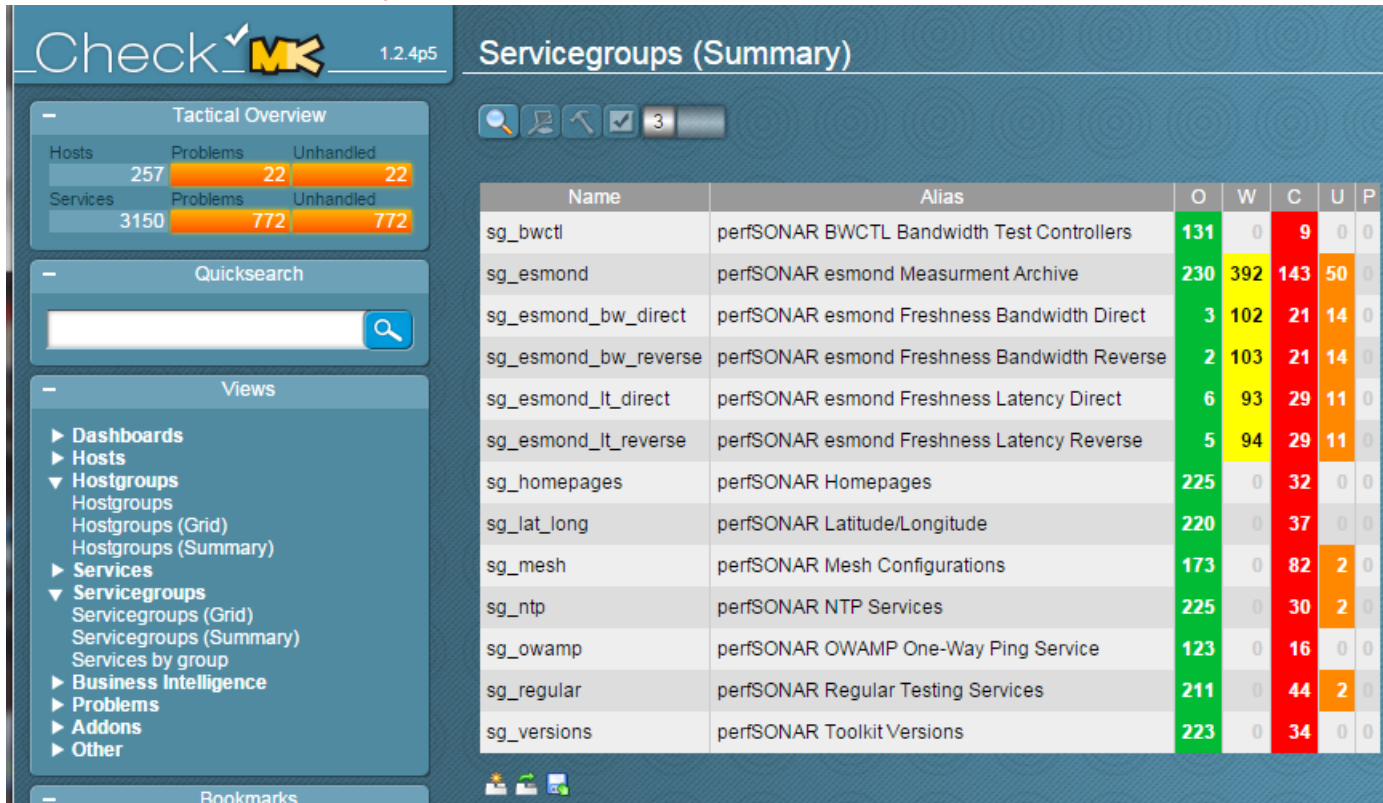
state	Host	Icons	Alias	OK	Wa	Un	Cr	Pd
UP	ccperfsonar1.in2p3.fr		ccperfsonar1.in2p3.fr	10	0	0	0	0
UP	ccperfsonar2.in2p3.fr		ccperfsonar2.in2p3.fr	10	0	0	0	0
UP	lapp-ps01.in2p3.fr		lapp-ps01.in2p3.fr	10	0	0	0	0
UP	lapp-ps02.in2p3.fr		lapp-ps02.in2p3.fr	10	0	0	0	0
UP	lhcone-wash-opt1.es.net		lhcone-wash-opt1.es.net	3	0	0	0	0
UP	lhcone.test.manlan.internet2.edu		lhcone.test.manlan.internet2.edu	11	0	0	0	0
UP	lhcone.test.wix.internet2.edu		lhcone.test.wix.internet2.edu	11	0	0	0	0
UP	perfsonar-de-kit.gridka.de		perfsonar-de-kit.gridka.de	10	0	0	0	0
UP	perfsonar-ps-01.desy.de		perfsonar-ps-01.desy.de	10	0	0	0	0
UP	perfsonar-ps-02.desy.de		perfsonar-ps-02.desy.de	10	0	0	0	0
UP	perfsonar.ultraight.org		perfsonar.ultraight.org	10	0	0	0	0
UP	perfsonar2-de-kit.gridka.de		perfsonar2-de-kit.gridka.de	10	0	0	0	0
UP	perfsonar2.ultraight.org		perfsonar2.ultraight.org	10	0	0	0	0
UP	ps-bandwidth.lhcmn.triumf.ca		ps-bandwidth.lhcmn.triumf.ca	10	0	0	0	0
UP	ps-bandwidth.scinet.utoronto.ca		ps-bandwidth.scinet.utoronto.ca	10	0	0	0	0
UP	ps-latency.lhcmn.triumf.ca		ps-latency.lhcmn.triumf.ca	10	0	0	0	0
UP	ps-latency.scinet.utoronto.ca		ps-latency.scinet.utoronto.ca	10	0	0	0	0



We are using OMD & Check_MK to monitor our perfSONAR hosts and services. Provides useful overview of status/problems

https://psomd.grid.iu.edu/WLCGperfSONAR/check_mk/

[Requires x509 in your browser]



Check_MK 1.2.4p5

Servicegroups (Summary)

Tactical Overview

Hosts	Problems	Unhandled
257	22	22
Services	Problems	Unhandled
3150	772	772

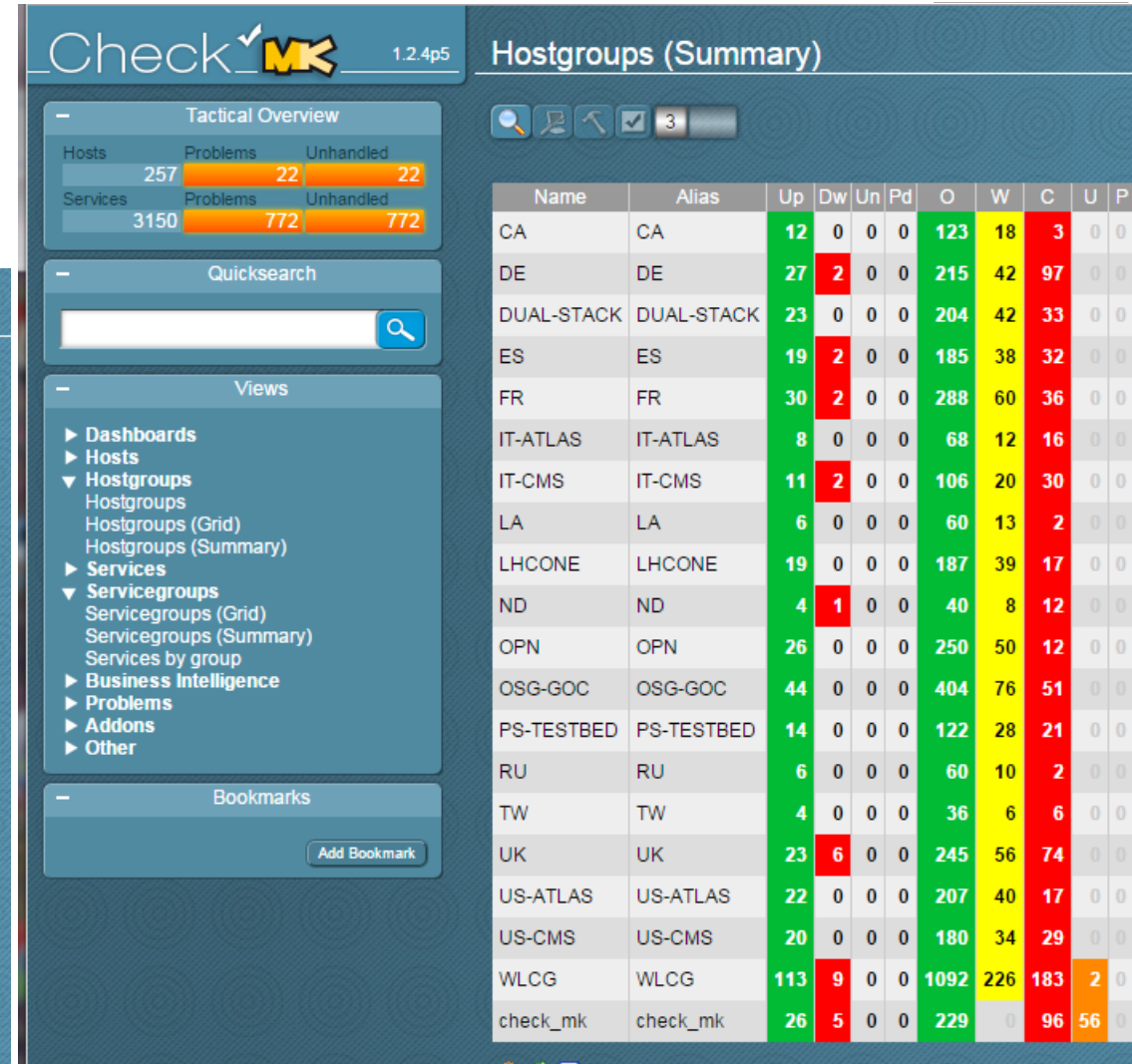
Quicksearch

Views

- ▶ Dashboards
- ▶ Hosts
- ▼ Hostgroups
 - Hostgroups
 - Hostgroups (Grid)
 - Hostgroups (Summary)
- ▶ Services
- ▼ Servicegroups
 - Servicegroups (Grid)
 - Servicegroups (Summary)
 - Services by group
- ▶ Business Intelligence
- ▶ Problems
- ▶ Addons
- ▶ Other

Bookmarks

Name	Alias	O	W	C	U	P
sg_bwctl	perfSONAR BWCTL Bandwidth Test Controllers	131	0	9	0	0
sg_esmond	perfSONAR esmond Measurement Archive	230	392	143	50	0
sg_esmond_bw_direct	perfSONAR esmond Freshness Bandwidth Direct	3	102	21	14	0
sg_esmond_bw_reverse	perfSONAR esmond Freshness Bandwidth Reverse	2	103	21	14	0
sg_esmond_it_direct	perfSONAR esmond Freshness Latency Direct	6	93	29	11	0
sg_esmond_it_reverse	perfSONAR esmond Freshness Latency Reverse	5	94	29	11	0
sg_homepages	perfSONAR Homepages	225	0	32	0	0
sg_lat_long	perfSONAR Latitude/Longitude	220	0	37	0	0
sg_mesh	perfSONAR Mesh Configurations	173	0	82	2	0
sg_ntp	perfSONAR NTP Services	225	0	30	2	0
sg_owamp	perfSONAR OWAMP One-Way Ping Service	123	0	16	0	0
sg_regular	perfSONAR Regular Testing Services	211	0	44	2	0
sg_versions	perfSONAR Toolkit Versions	223	0	34	0	0



Check_MK 1.2.4p5

Hostgroups (Summary)

Tactical Overview

Hosts	Problems	Unhandled
257	22	22
Services	Problems	Unhandled
3150	772	772

Quicksearch

Views

- ▶ Dashboards
- ▶ Hosts
- ▼ Hostgroups
 - Hostgroups
 - Hostgroups (Grid)
 - Hostgroups (Summary)
- ▶ Services
- ▼ Servicegroups
 - Servicegroups (Grid)
 - Servicegroups (Summary)
 - Services by group
- ▶ Business Intelligence
- ▶ Problems
- ▶ Addons
- ▶ Other

Bookmarks

Add Bookmark

Name	Alias	Up	Dw	Un	Pd	O	W	C	U	P
CA	CA	12	0	0	0	123	18	3	0	0
DE	DE	27	2	0	0	215	42	97	0	0
DUAL-STACK	DUAL-STACK	23	0	0	0	204	42	33	0	0
ES	ES	19	2	0	0	185	38	32	0	0
FR	FR	30	2	0	0	288	60	36	0	0
IT-ATLAS	IT-ATLAS	8	0	0	0	68	12	16	0	0
IT-CMS	IT-CMS	11	2	0	0	106	20	30	0	0
LA	LA	6	0	0	0	60	13	2	0	0
LHCONE	LHCONE	19	0	0	0	187	39	17	0	0
ND	ND	4	1	0	0	40	8	12	0	0
OPN	OPN	26	0	0	0	250	50	12	0	0
OSG-GOC	OSG-GOC	44	0	0	0	404	76	51	0	0
PS-TESTBED	PS-TESTBED	14	0	0	0	122	28	21	0	0
RU	RU	6	0	0	0	60	10	2	0	0
TW	TW	4	0	0	0	36	6	6	0	0
UK	UK	23	6	0	0	245	56	74	0	0
US-ATLAS	US-ATLAS	22	0	0	0	207	40	17	0	0
US-CMS	US-CMS	20	0	0	0	180	34	29	0	0
WLCG	WLCG	113	9	0	0	1092	226	183	2	0
check_mk	check_mk	26	5	0	0	229	0	96	56	0

MaDDash for Network Metric Visualization

Visibility of our perfSONAR metrics are being exposed using Esnet's MaDDash (Monitoring and Debugging Dashboard) which presents a quick way to see the status of our network measurements.

The colors indicate status (based upon configured levels) where "OK" is green, "WARNING" is yellow and "CRITICAL" is red.

What we **don't** want to see is orange which indicates the measurement data is not available.

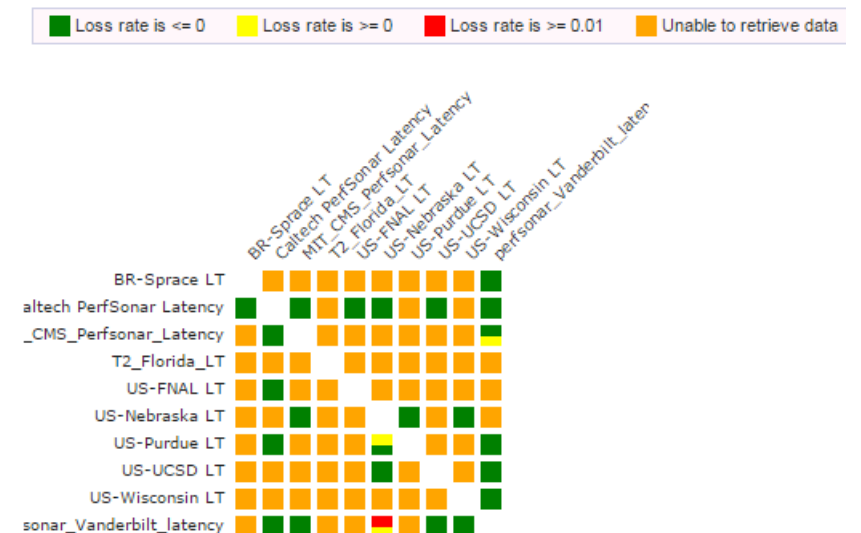
On the right are the US CMS bandwidth and latency meshes as of today: <http://psmad.grid.iu.edu/maddash-webui/index.cgi?dashboard=USCMS%20Mesh%20Config>

For debugging examples see my talk from APAN August 2014: http://www.apan.net/meetings/Nantou2014/Sessions/LHCONE/LHCO NE_perfSONAR_status-APAN.pdf

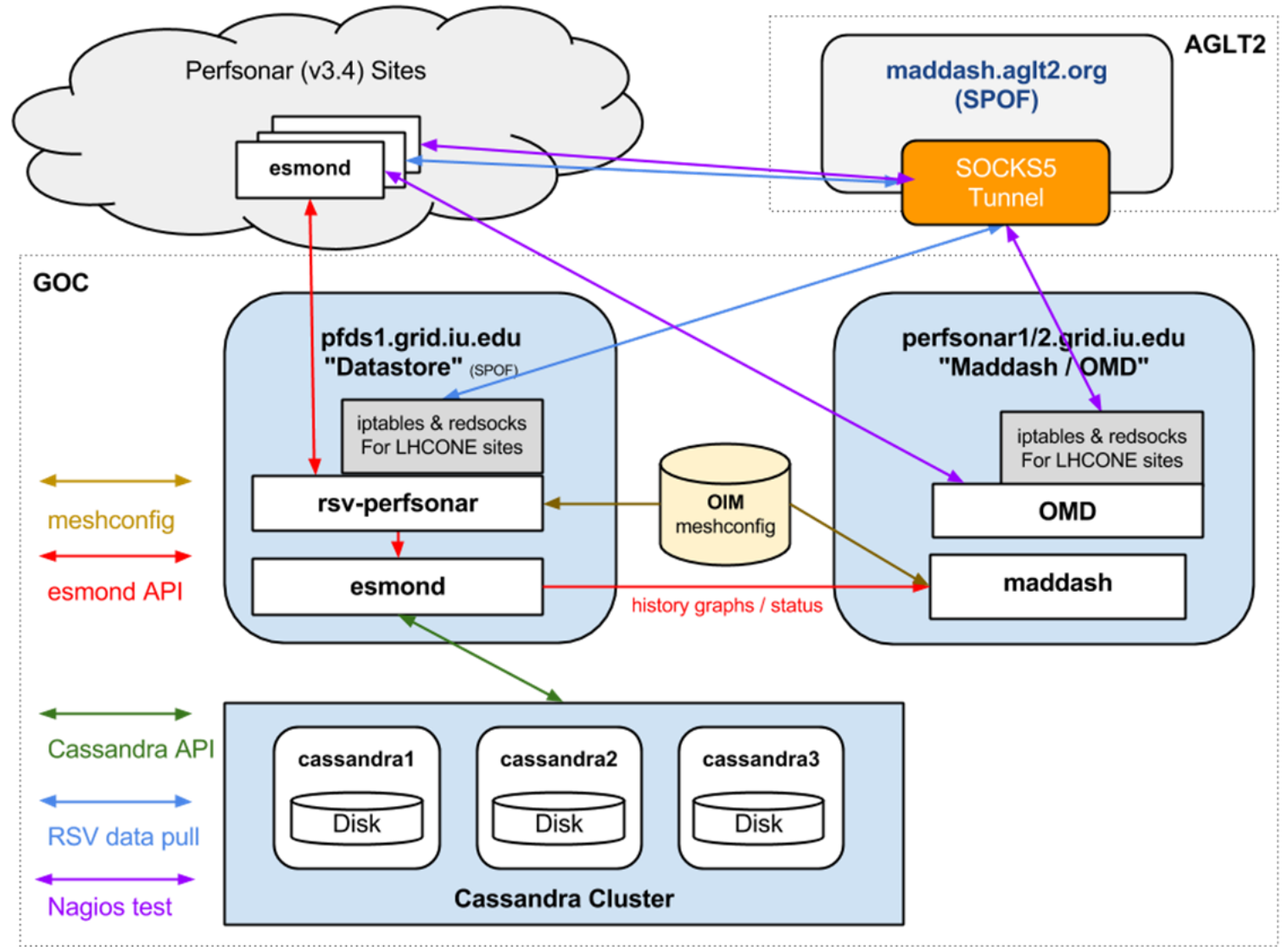
USCMS Mesh Config - USCMS Bandwidth Mesh Test



USCMS Mesh Config - USCMS Latency Mesh Test



- ❄️ A critical component is the datastore to organize and store the network metrics and associated metadata
 - ❑ OSG has begun gathering relevant metrics from the complete set of OSG and WLCG perfSONAR instances
 - ❑ This data will be available via an API, must be visualized and must be organized to provide the “OSG Networking Service”
 - ❑ Targeting a production service by mid-summer
 - ❑ Examples at end of slides



Near Term Goals: Where Are We Going?

- ❄ We are working on a number of improvements. Perhaps the most important one should be released this week: perfSONAR 3.4.2. Everyone who installed 3.4 should “automatically” update within 24-hours (unless you turned it off ☹)
- ❄ The OSG Network Datastore as our repository of network measurements is targeting a production-ready date of July 2015. This is planned to be a permanent archive of all WLCG/OSG perfSONAR metrics going forward.
- ❄ Improvements in MaDDash data presentation (linking labels back to toolkit instances, more customization options, bug-fixes)
- ❄ Development of PuNDIT agents that use perfSONAR measurements to identify (in near real-time) network problems and a corresponding central service to localize where those problems are coming from.
- ❄ Longer term improvements in perfSONAR regarding network topology and in support problem characterization and reporting

❄ I wanted to reserve time for an open discussion about your experiences with perfSONAR

- ❑ What issues do you have?
- ❑ What would you like to know more about?
- ❑ Do you know where to go to get information?

❄ Questions, comments, suggestions?

Floor is open....



Further reference

ADDITIONAL SLIDES

Relevant URLs For More Information

- ❄ OSG/WLCG documentation “tree” at <https://twiki.opensciencegrid.org/bin/view/Documentation/DeployperfSONAR>
- ❄ perfSONAR documentation at <http://docs.perfsonar.net/>
- ❄ Troubleshooting information at <http://fasterdata.es.net/performance-testing/network-troubleshooting-quick-reference-guide/>
- ❄ OMD/Check_MK (use a browser with your x509 cert!): https://psomd.grid.iu.edu/WLCGperfSONAR/check_mk/index.py?start_url=%2FWLCGperfSONAR%2Fcheck_mk%2Fdashboard.py
- ❄ MaDDash: (prototype) <https://maddash.aglt2.org/maddash-webui/index.cgi?dashboard=USCMS%20Mesh%20Config> (production; relies upon datastore) <http://psmad.grid.iu.edu/maddash-webui/index.cgi?dashboard=USCMS%20Mesh%20Config>
- ❄ PuNDIT project <http://pundit.gatech.edu/>

perfSONAR Global Deployment Trend

As of February 2015 there are almost 1300 perfSONAR deployments worldwide.

The OSG/LHC deployment accounts for 247 of those.

The figure on the right shows the deployment count by version of perfSONAR along with some events highlighted in time via the red arrows.

Security issues in the underlying OS for perfSONAR helped motivate the upgrade to v3.4

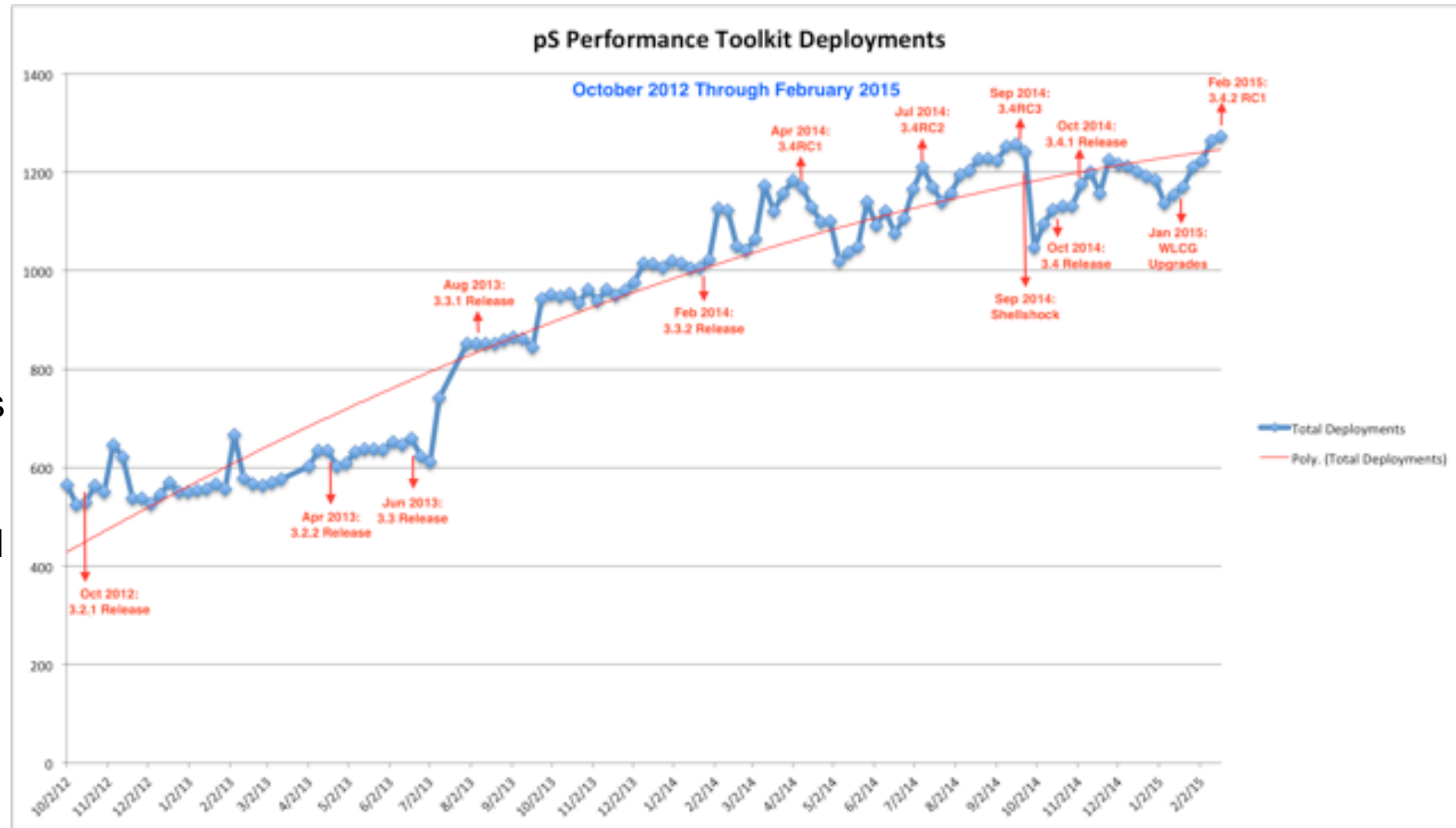


Diagram courtesy of Jason Zurawski/ESnet

OSG Datastore Details

* esmond – Postgress + Cassandra

- populated by RSV probes

* REST API available – python/perl libs

- `curl "http://archive.example.net/esmond/perfsonar/archive/fce0483e51de49aaa7fcf8884d053134/histogram-owdelay/base?time-range=86400"`

* Data organized in events

- [packet-trace](#)
- [histogram-owdelay](#) – one way delays over time period
- [ntp-delay](#) – round trip delay time to NTP server
- [packet-loss-rate](#) – number of packets lost/packets sent
- [packet-count-sent](#) – packets sent
- [packet-count-lost](#) – packets lost
- [packet-retransmits](#) – packets retransmitted for a transfer using TCP
- [throughput](#) – observer amount of data sent over period of time
- [failures](#) – record of test failures

Datastore Structure

```
[
  {
    "source": "10.1.1.1",
    "destination": "10.1.1.2",
    "event-types": [
      {
        "base-uri": "/esmond/perfsonar/archive/f6b732e9f351487a96126f0c25e5e546/packet-retransmits/base",
        "event-type": "packet-retransmits",
        "summaries": [
          ],
        "time-updated": 1397482734
      },
      {
        "base-uri": "/esmond/perfsonar/archive/f6b732e9f351487a96126f0c25e5e546/throughput/base",
        "event-type": "throughput",
        "summaries": [
          {
            "summary-type": "average",
            "summary-window": "86400",
            "time-updated": 1397482735,
            "uri": "/esmond/perfsonar/archive/f6b732e9f351487a96126f0c25e5e546/throughput/averages/86400"
          }
        ],
        "time-updated": 1397482735
      },
    ],
    "input-source": "host1.example.net",
    "input-destination": "host2.example.net",
    "ip-transport-protocol": "tcp",
    "measurement-agent": "10.1.1.1",
    "metadata-key": "f6b732e9f351487a96126f0c25e5e546",
    "subject-type": "point-to-point",
    "time-duration": "20",
    "time-duration": "14400",
    "tool-name": "bwctl/iperf3",
    "uri": "/esmond/perfsonar/archive/f6b732e9f351487a96126f0c25e5e546/"
  }
]
```

Data Examples: Throughput vs OWdelay vs Trace

```
[
  {
    "ts":1397421672,
    "val":7016320000.0
  },
  {
    "ts":1397442692,
    "val":7225480000.0
  },
  {
    "ts":1397466492,
    "val":7095460000.0
  },
  {
    "ts":1397482700,
    "val":7042540000.0
  }
]
```

```
[
  {
    "ts":1397504013,
    "val":{
      "34.4":506,
      "34.5":85,
      "34.6":5,
      "34.7":4
    }
  },
  {
    "ts":1397504052,
    "val":{
      "34.4":510,
      "34.5":80,
      "34.6":7,
      "34.7":3
    }
  },
  .....
]
```

```
{
  "ts":1397566094,
  "val":[
    {
      "error_message":null,
      "ip":"198.124.238.65",
      "mtu":"9000",
      "query":"1",
      "rtt":"0.246",
      "success":1,
      "ttl":"1"
    },
    {
      "error_message":null,
      "ip":"198.124.238.65",
      "mtu":"9000",
      "query":"2",
      "rtt":"0.195",
      "success":1,
      "ttl":"1"
    }
  ],
}
```