

# Deployment and Testing of MaDDash and OMD for WLCG perfSONAR-PS Monitoring

This page documents testing and deploying MaDDash and OMD (Open Monitoring Distribution; see <http://omdistro.org/start>) for WLCG perfSONAR-PS monitoring. We need to determine how this system might work for OSG and WLCG needs.

## Initial Deployment

To test the deployment I am using a new SL6.4 64-bit VM created on our AGLT2 VMware system. The system was setup with 1 processor, 4GB of RAM and 64GB of disk. The OS in VMware is CentOS 64-bit (/4/5/6). I used our local provisioning system to do the "bare-metal" install of the OS. Others can use whatever infrastructure they have which does the equivalent.

Once the system is built I add in three yum repositories:

1) Make sure EPEL is installed. `rpm -Uvh`

```
"http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm"
```

2) Then to get OMD use `rpm -Uvh`

```
"http://labs.consol.de/repo/stable/rhel6/i386/labs-consol-stable.rhel6.noarch.rpm"
```

3) Get the perfSONAR-PS repository in place (I am using 3.3.2 RC): `rpm -Uvh`

```
"http://software.internet2.edu/branches/release-3.3.2/rpms/el6/x86_64/RPMS.main/Internet2-repo-0."
```

It may be a good idea to run `yum update` to make sure you are current. FYI my repolist ended up like:

```
[root@maddash ~]# yum repolist
Loaded plugins: security
repo id                                repo name
Internet2                              Internet2 RPM Repository - software.internet2.edu - main
epel                                    Extra Packages for Enterprise Linux 6 - x86_64
labs_consol_stable                     labs_consol_stable
rpmforge                                RHEL 6.4 - RPMforge.net (formerly dag)
sl                                      Scientific Linux 6.4 - x86_64
sl-security                             SL 6.4 security updates
umatlas                                 umatlas SL 6.4
vmware-tools                           vmware-tools-collection
repolist: 22,102
```

## Install

To install Maddash I followed the instructions at

<https://code.google.com/p/perfsonar-ps/wiki/MaDDashInstall>.

```
yum install maddash
```

NOTE: This pulled in 81 packages on my system.

## Install OMD

To install OMD : `yum install omd-1.10`

NOTE: This pulled in 25 packages on my system.

## Next Steps

We eventually need to setup and configure MaDDash and OMD for our use-case.

The idea is to use OMD to monitor the perfSONAR-PS Toolkit nodes and basic services We will setup host-groups based upon cloud and/or VO.

MaDDash will be used to visualize network measurements being made by the perfSONAR-PS Toolkits.

In both cases we need to get the lists of nodes and corresponding metadata for the WLCG perfSONAR-PS monitoring. Fortunately we have the "mesh-configs" already in place which should provide us with the correct information. The list of mesh-configs is stored at <https://twiki.cern.ch/twiki/bin/view/LCG/MeshRegionList> We need to be able to parse these to extract the relevant details.

## Setup OMD

We want to setup OMD and create a new 'site' but first we need to fix a known bug on RHEL-like systems. From <http://everyday-tech.com/archives/1999>:

"In CentOS 6.4 there is a small issue with pathing in /usr/bin/omd. On line 794 You want to add the following highlighted text .

```
file( /etc/fstab , a+ ).write( tmpfs */opt*%s tmpfs noauto,user,mode=755,uid=%s,gid=%s 0 0\n  % \"
```

I edited /usr/bin/omd and added the missing /opt.

Now we can setup omd with `omd setup`. This adds and configures some packages. Note it chkconfigs httpd on. If you have some security in place you may need to configure to allow httpd to be accessible as required.

Once OMD is setup we can create a new 'site'. I choose the sitename 'WLCGperfSONAR'. We can run 'omd create' now:

```
[root@maddash ~]# omd create WLCGperfSONAR
Adding /omd/sites/WLCGperfSONAR/tmp to /etc/fstab.
Restarting Apache...OK
Creating temporary filesystem /omd/sites/WLCGperfSONAR/tmp...OK
Created new site !WLCGperfSONAR with version 1.10.
```

```
The site can be started with omd start WLCGperfSONAR.
The default web UI is available at http://maddash.aglt2.org/WLCGperfSONAR/
The admin user for the web applications is omdadmin with password omd.
Please do a su - !WLCGperfSONAR for administration of this site.
```

So we are setup and ready to run. We should first start the site, login and change the default password. Here we hit another snag (at least on my system):

```
[root@maddash ~]# omd start WLCGperfSONAR
Starting dedicated Apache for site WLCGperfSONAR...OK
Starting rrdcached...OK
Starting npcd...OK
Starting nagios...OK
Initializing Crontab...You (WLCGperfSONAR) are not allowed to use this program (/usr/bin/crontab)
See crontab(1) for more information
close failed in file object destructor:
Error in sys.excepthook:

Original exception was:
ERROR
```

We need to add the new 'WLCGperfSONAR' user to the /etc/cron.allow on this system and retry:

```
[root@maddash ~]# omd start WLCGperfSONAR
Starting dedicated Apache for site WLCGperfSONAR...OK
Starting rrdcached...OK
Starting npcd...OK
Starting nagios...OK
Initializing Crontab...OK
```

Worked! We login via the web page and change the omdadmin password. Easiest way to do this is to open your new site URL <http://maddash.aglt2.org/WLCGperfSONAR/> If you have allowed http access via any firewall it should prompt for a user/password. Use 'omdadmin' and 'omd'. It will take you to the main site page showing lots of applications you can use. Click on the "Check\_MK" version and go to the 'WATO Configuration' on the left panel. Select 'Users and Contacts' and then you can click on the green pencil icon ("Properties") next to the 'omdadmin' account and setup a new password. It will immediately prompt you to re-login with the new password.

## OMD WLCGperfSONAR Configuration

To configure the site you could use the WATO interface on the web or you can login and use command line tools. WATO may be an option we can explore later. For now, login as root on the new system and `su - WLCGperfSONAR` to become the new site owner/user. The "root" of the site is in /omd/sites/ which is /omd/sites/WLCGperfSONAR in our case.

The easiest way to use/configure OMD is to exploit **check\_mk**. The configuration files are in /omd/sites/WLCGperfSONAR/etc/check\_mk and sub-directories. The main file is appropriately called **main.mk**. Any files ending in .mk will be included. The 'wato' subdirectories are for WATO (Web Administration Tool). One requirement is that the ../check\_mk/conf.d/wato/hosts.mk file must be the one which holds the systems to add or they won't show up in the web-admin screen.

I have created some perl scripts in the /omd/sites/WLCGperfSONAR/etc/check\_mk/conf.d/wato directory to help configure things for WLCG use.

- geoaddress.pl --- Subroutine to find "standard" address from location description
- geocode.pl --- Subroutine using Google map API to find latitude/longitude from address
- test-geocode.pl --- Test program for geocode subroutine
- get\_ps\_info.pl --- This is a subroutine which loads perfSONAR-PS lookup information into a perl hash for use by the caller
- test-get\_ps\_info.pl --- This script call the above subroutine. Call it with the Fully Qualified Domain Name (FQDN) of the perfSONAR-PS host you want information on:
  - ◆ perl test-get\_ps\_info.pl psum01.aglt2.org
- parse-mesh-url.pl --- An example of parsing **WLCG perfSONAR-PS** meshes (JSON files).
- test-scrape.pl --- Test perl code to "screen-scrape" PS Toolkit information
- wlcg-mesh-to-wato.pl --- This routine creates a number of .mk files (for **Check\_MK** and **WATO**) from the WLCG mesh information. One file hosts-add.mk will need to be integrated into the local hosts.mk file. This script finds all WLCG perfSONAR-PS hosts and creates tags for service type (owamp or bwctl), creates host-groups by cloud and service type and extracts latitude and longitude for Nagvis.

In addition we have some specific **Check\_mk** configuration files that we need:

- ntp.mk --- Check\_mk configuration for NTP parameters
- extra\_nagios\_conf.mk --- Check\_mk configuration for Nagios legacy port check and web page commands
- legacy\_checks.mk --- Check\_mk legacy checks for perfSONAR-PS monitoring

The test instance is available at <https://maddash.aglt2.org/WLCGperfSONAR/omd/>. The readonly user is 'WLCGps'. Ask if you need the password (smckee 'at' umich.edu). See at the bottom for some additional information about interesting things to check out.

## MaDDash Installation

MaDDash is easy to install and is documented at <https://code.google.com/p/perfsonar-ps/wiki/MaDDashInstall>. The quickstart section shows:

1. Login to a host running the perfSONAR-PS Toolkit 3.2.2 or later.
2. Run the following command as a privileged user to install the software: `# yum install maddash`
3. Open the file `/etc/maddash/maddash-server/maddash.yaml` and change the following properties (Note: Use spaces and not tabs in this file. YAML does not allow tabs.)
4. Under the groups section, change the myOwampHosts list and the myBwctlHosts list to the list of OWAMP and BWCTL hosts you wish to check, respectively. NOTE: If you comment out one of the groups because you don't want any BWCTL or OWAMP checks, then also remove the corresponding entry under the "grids" section of the file. Do a search and replace for example.mydomain.local and change it to the hostname of the toolkit host on which the software is installed. This information will be used to generate the graphs.
5. Restart the server: `# /etc/init.d/maddash-server restart`
6. Open the maddash web page in your browser at the following URL (replace MYHOST with the name of your host): <http://MYHOST/maddash-webui>

You should now be able to view the results of the checks being run. See the remainder of this document for more detailed customization features.

The test instances is at <http://maddash.aglt2.org/maddash-webui>

## Configuring MaDDash

We are interested in using MaDDash to monitor our WLCG perfSONAR-PS clouds and their corresponding tests. There is a section of the MaDDash documentation that covers this at [https://code.google.com/p/perfsonar-ps/wiki/MaDDashInstall#Advanced\\_Topic:\\_Using\\_the\\_perfSONAR\\_Mesh\\_Conf](https://code.google.com/p/perfsonar-ps/wiki/MaDDashInstall#Advanced_Topic:_Using_the_perfSONAR_Mesh_Conf). The quickstart is:

1. Login to the command-line interface of your host running MaDDash via SSH or a local terminal
2. Install the MeshConfig software `# yum install perl-perfSONAR_PS-MeshConfig-GUIAgent`
3. Update `/opt/perfsonar_ps/mesh_config/etc/gui_agent_configuration.conf` with the URL of your JSON file

```
<mesh>
    configuration_url           https://host.domain.edu/example.json
</mesh>
...
```

1. Run the following command to generate your maddash configuration for the first time `# /opt/perfsonar_ps/mesh_config/bin/generate_gui_configuration`
2. Go to your MaDDash web interface to verify the results

You should now have a MaDDash configuration that will update nightly based on the published mesh

I found I additionally needed to install the `perl-perfSONAR_PS-Toolkit` rpm via `yum install perl-perfSONAR_PS-Toolkit` or I got errors when I tried to actually run the `/opt/perfsonar_ps/mesh_config/bin/generate_gui_configuration` command (nothing was listening on port 9000 otherwise). The needed service is **config\_daemon**.

I tried to use the WLCG mesh configurations listed here <https://twiki.cern.ch/twiki/bin/view/LCG/MeshRegionList>. I ran into problems trying to generate the configuration on all but three meshes: USCMS, USATLAS and FR. The two types of errors are shown below.

For the DE, ES, LHCOPN and WLCG meshes:

```
#2013/12/27 16:51:59 (972464) ERROR> GUIAgent.pm:282 perfSONAR_PS::MeshConfig::GUIAgent::__genera
#2013/12/27 16:51:59 (972464) ERROR> GUIAgent.pm:228 perfSONAR_PS::MeshConfig::GUIAgent::__config
```

For the ITCMS, ITATLAS, UK and RU meshes

```
#2013/12/27 16:57:07 (977880) ERROR> GUIAgent.pm:282 perfSONAR_PS::MeshConfig::GUIAgent::__genera
#2013/12/27 16:57:07 (977880) ERROR> GUIAgent.pm:228 perfSONAR_PS::MeshConfig::GUIAgent::__config
```

The TW, CA, CERN and NL meshes are not configured.

I tried to update the USATLAS mesh to include the new network address for the MWT2\_UC nodes but this change seems to have broken the USATLAS mesh which now gives:

```
2014/01/03 09:31:15 (2216221) ERROR> GUIAgent.pm:282 perfSONAR_PS::MeshConfig::GUIAgent::__genera
2014/01/03 09:31:15 (2216221) ERROR> GUIAgent.pm:228 perfSONAR_PS::MeshConfig::GUIAgent::__config
```

**Aaron Brown provided a quick, patch to let me find the problematic hosts. There were cases in some meshes where Latency hosts were in the Bandwidth mesh and vice-versa. These are mostly fixed now and problems remain only with the ES cloud-mesh and the whole WLCG mesh.** We still need to create meshes for CA, CERN, ND, NL and TW.

## Plans and Areas to Explore

There are a number of areas to explore for both OMD and MaDDash.

For MaDDash:

- We need to resolve the existing mesh-errors
- We need to verify the scalability when all meshes are operating. With 3 meshes (USATLAS, USCMS and FR) we have seen loads spike to 10 on the VM
- Mesh names are truncated on some browsers and this needs fixing
- It would be useful to hyperlink the row and column names to the corresponding perfSONAR-PS toolkit web pages

For OMD:

- We have scripts that parse the meshes and create a basic Check\_MK/WATO configuration. These could be improved or extended.
- For each node we setup a basic 'ping' and 'http' check. The 'http' check tries to access the perfSONAR-PS toolkit page on each node. Better test configurations may be possible.
- Depending upon the type of node (Latency or Bandwidth) we test specific HTTP ports for services that should be running. We should explore more service oriented testing using the ESnet plugins.
- If the nodes are running the check\_mk-agent we gather significantly more data (See the **psum01.aglt2.org** and **psum02.aglt2.org** instances as an example). To make this easier we should:
  - ◆ Put the needed RPM(s) for the check\_mk-agent in a well-known repo so sites can install it.
  - ◆ Sites need to enable port 6556 for access from the OMD WLCG monitoring instance.
- We should explore specific tests we can implement like:
  - ◆ Test for PS Version and WARN if one version back and CRITICAL if older

## MadDashWLCG < LCG < TWiki

- ◆ Test for Global Registration (to verify instances are registered correctly). I noticed a number of the WLCG instances are either not found in the lookup service or have minimal information.
- ◆ Test for existence of latitude and longitude numbers
- ◆ Test for admin name and email (NOTE: Lookup service doesn't seem to return this....Bug?)
- We need to setup appropriate contacts for each host and cloud in OMD to allow customized alerting (when we are ready)
- General Nagios-like configuration improvements.

I am sure there are lots of other things to explore.

-- ShawnMcKee - 30 Dec 2013

---

This topic: LCG > MadDashWLCG

Topic revision: r8 - 2014-02-28 - ShawnMcKee



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.  
Ideas, requests, problems regarding TWiki? Send feedback