

Performance Benchmark Information Provider

Performance Benchmark Information Provider

Abstract

The basic idea of this Information Provider is to allow a set of performance benchmarks to be run and to aggregate the results of the test back into the WS MDS Index Service. To do this, each test must be configured separately, and each set of results will be placed in a separate Resource Property (RP). This simplifies the configuration and allows for easy client side querying of the results of an individual test. This information provider also includes a "Hello, World!" test that you should use alone to make sure that your environment is configured properly so that the other (more complicated) tests might have a chance of working as well.

You can download a PDF version of Performance Benchmark Information Provider information [here](#)¹.

¹ perf_benchmark.pdf

Table of Contents

1. Release Notes	1
1. Component Overview	1
2. Feature Summary	1
3. Changes Summary	1
4. Bug Fixes	1
5. Known Problems	1
6. Technology Dependencies	1
7. Tested Platforms	2
8. Backward Compatibility Summary	2
9. Associated Standards	2
10. For More Information	2
2. Reference Guide	3
1. Overview	3
2. Prerequisites	3
3. Configuring	4
4. Resource Properties	21
5. Schema	21
6. Security Considerations	21
7. Testing	21
8. Troubleshooting	21
Glossary	22

Chapter 1. GT 4.2.1 Release Notes: Performance Benchmark Information Provider

1. Component Overview

The basic idea of this Information Provider is to allow a set of performance benchmarks to be run and to aggregate the results of the test back into the WS MDS Index Service. To do this, each test must be configured separately, and each set of results will be placed in a separate Resource Property (RP). This simplifies the configuration and allows for easy client side querying of the results of an individual test. This information provider also includes a "Hello, World!" test that you should use alone to make sure that your environment is configured properly so that the other (more complicated) tests might have a chance of working as well.

2. Feature Summary

- [TODO: list new features of this info provider for this release]

3. Changes Summary

This is a new information provider for GT 4.2.

4. Bug Fixes

There are no fixed bugs for this information provider.

5. Known Problems

The following problems and limitations are known to exist for WS MDS GKrellm Service at the time of the 4.2.1 release:

5.1. Limitations

- [list]

5.2. Known Bugs

- There are currently no bugs for this information provider.

6. Technology Dependencies

The Index Service depends on the following GT components:

- [Java WS Core](#)

The Index Service depends on the following 3rd party software:

- All of the systems that the user wishes to observe via this information provider must be running gkrellmd version 2.2.7 or greater. More information on gkrellmd can be found at: <http://www.gkrellm.net/>

7. Tested Platforms

Tested Platforms for WS-MDS Index Service:

- Linux on i386
- Linux on x86_64

Tested containers for this Information Provider

- Java WS Core container

8. Backward Compatibility Summary

This is a new information provider with this version.

9. Associated Standards

Associated standards for this Information Provider:

- [list]

10. For More Information

See [Chapter 2, GT 4.2.1: Performance Benchmark Information Provider Reference](#) for more information about this information provider.

Chapter 2. GT 4.2.1: Performance Benchmark Information Provider Reference

1. Overview

The basic idea of this Information Provider is to allow a set of performance benchmarks to be run and to aggregate the results of the test back into the WS MDS Index Service. To do this, each test must be configured separately, and each set of results will be placed in a separate Resource Property (RP). This simplifies the configuration and allows for easy client side querying of the results of an individual test. This information provider also includes a "Hello, World!" test that you should use alone to make sure that your environment is configured properly so that the other (more complicated) tests might have a chance of working as well.

REQUIRED: indicate which module this provider is implemented with (usefulrp or aggregator) and add link.

2. Prerequisites

1. A cluster that has a shared file system between the login (or job submission) nodes and the backend (compute) nodes. For example, all backend (compute) nodes must be able to find a common file on some (arbitrary) mount point such as /nfs/shared/foo. As long as the program and output file can be run and written, respectively, there should be no trouble using this software.
2. A working GT4 installation with a configured WS-GRAM installation so that jobs can be submitted (via PBS, or Condor). A single running container is required for this information provider, and the job is submitted to this container.

To satisfy this requirement, you should be able to have a container running on the login node and be able to run a command such as the following on the command line without error:

```
$GLOBUS_LOCATION/bin/globusrun-ws -submit -Ft PBS -F \  
  https://MYHOST:MYPORT/wsrp/services/ManagedJobFactoryService \  
  -c /bin/true
```

The output should look similar to this:

```
-----  
Submitting job...Done.  
Job ID: uuid:7790abec-e5d9-11da-b93a-0014221d2259  
Termination time: 05/18/2006 19:15 GMT  
Current job state: Pending  
Current job state: Active  
Current job state: CleanUp  
Current job state: Done  
Destroying job...Done.  
-----
```

If this does not work without error on your system (where PBS should have been substituted for the scheduler installed on your system), please consult your system administrator or the WS-GRAM documentation located at: <http://www-unix.globus.org/toolkit/docs/4.0/execution/wsgram/user-index.html>

- 3) A working and configured MPI installation on the backend nodes.
- 4) An installation of the following programs that are accessible (i.e. can be run) from the backend (compute) nodes:

MPPTTEST: <http://www-unix.mcs.anl.gov/mpi/mpptest/>

MPIBENCH: <http://icl.cs.utk.edu/projects/llcbench/mpbench.html>

LLCBENCH: <http://icl.cs.utk.edu/projects/llcbench/index.html>

STREAM: <http://www.cs.virginia.edu/stream/>

3. Configuring

The following configuration is required for this information provider:

1. You must have the `$GLOBUS_LOCATION` and `$MPI_LOCATION` environment variables set. The `GLOBUS_LOCATION` must point to the root of a valid GT4 installation (such as `/nfs/software/globus-4.0.2`). Similarly, the `MPI_LOCATION` variable must point to the root of a valid MPI installation (such as `/nfs/software/mpich-1.2.7`). If you're unsure how to set these environment variables, check with your site administrator. Common ways to do this are like this:

```
[ bash users, try this ]
```

```
bash # export GLOBUS_LOCATION=/nfs/software/globus-4.0.2
bash # export MPI_LOCATION=/nfs/software/mpich-1.2.7
```

```
[ tcsh users, try this ]
```

```
tcsh $ setenv GLOBUS_LOCATION /nfs/software/globus-4.0.2
tcsh $ setenv MPI_LOCATION /nfs/software/mpich-1.2.7
```

2. The first major step is to make sure that your GT4 installation is a recent enough version and is compatible with this Information Provider. If you're running GT 4.0.2, you will need to update your installation with the May 5 WS-MDS RPPProvider patch located at:

<http://www.globus.org/toolkit/downloads/development/>

Install this patch by setting your `GLOBUS_LOCATION` to the appropriate directory and then running the following commands:

```
$GLOBUS_LOCATION/sbin/gpt-build -update gt4.0.2-wsmads-update-1.0-src_bundle.tar.gz
$GLOBUS_LOCATION/sbin/gpt-postinstall -force
```

3. When this step is complete, you should now have a file located at `$GLOBUS_LOCATION/etc/globus_wsrf_mds_usefulrp/gluece-rpprovider-sample-config.xml`. Copy this file to `$GLOBUS_LOCATION/etc/globus_wsrf_mds_usefulrp/rp-provider-config.xml`.

4. Edit your `$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/server-config.wsdd` file.

Locate the `DefaultIndexService` section at the very bottom and replace it with the following:

```

<service name="DefaultIndexService" provider="Handler"
  use="literal" style="document">
  <parameter name="providers"
    value="org.globus.wsrfl.impl.servicegroup.ServiceGroupRegistrationProvider
    org.globus.mds.usefulrp.rpprovider.ResourcePropertyProviderCollection
    GetRPPProvider
    GetMRPPProvider
    QueryRPPProvider
    DestroyProvider
    SetTerminationTimeProvider
    SubscribeProvider
    GetCurrentMessageProvider" />

  <parameter name="handlerClass"
    value="org.globus.axis.providers.RPCProvider" />
  <parameter name="scope" value="Application" />
  <parameter name="allowedMethods" value="*" />
  <parameter name="rpProviderConfigFile"
    value="/etc/globus_wsrfl_mds_usefulrp/rp-provider-config.xml" />
  <parameter name="className"
    value="org.globus.mds.index.impl.DefaultIndexService" />
  <wsdlFile>share/schema/mds/index/index_service.wsdl</wsdlFile>
</service>

```

- Download the [exec_wrapper script](#)¹ and the [perf_benchmark script](#)² to a location on your cluster that is accessible to the login node as well as the backend (compute) nodes. While the perf_benchmark is only run on the login node, the exec_wrapper must be accessible because it will be run on each of the backend nodes.

After placing these files in a suitable shared location, please make sure that they are executable by issuing the following shell command:

```
chmod a+x exec_wrapper perf_benchmark
```

- Finally, the rest of the configuration will be a matter of editing your `$GLOBUS_LOCATION/etc/globus_wsrfl_mds_usefulrp/rp-provider-config.xml` file. Assuming that the Cluster Monitoring and Scheduling data is already properly configured (which is outside the scope of this document), we need to make some edits.

3.1. Configuring the Information Provider to Run the HELLO WORLD performance test

The following block of XML is the configuration that is needed to enable the "Hello, World!" test in the Index Service. It uses the RPPProvider framework to create a Resource Property (RP) called PerfBM-HelloWorld that will contain the hello world information upon successful execution.



Note

Note that this configuration block must be added to the `$GLOBUS_LOCATION/etc/globus_wsrfl_mds_usefulrp/rp-provider-config.xml` file.

¹ exec_wrapper

² perf_benchmark

```

<ns1:resourcePropertyName xsi:type="xsd:QName"
  xmlns:perfbm="http://perfbm-testing">perfbm:PerfBM>HelloWorld</ns1:resourcePro
<ns1:resourcePropertyImpl
  xsi:type="xsd:string">org.globus.mds.usefulrp.rprovider.SingleValueResourcePr
<ns1:resourcePropertyElementProducers
  xsi:type="ns1:resourcePropertyElementProducerConfig">
<ns1:className xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">org.globus.mds.usefulrp.rprovide

<!-- *** SPECIFY THE SCRIPT TO RUN HERE *** -->

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/perf_benchmark<

<!-- *** BEGIN SCRIPT ARGUMENTS *** -->

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">SCHEDULER</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">https://MYHOST:MYPORT/wsrf/servic

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">4</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/tmp</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/exec_wrapper</n

<ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">hello_world</ns1:arguments>


<!-- *** END ARGUMENTS *** -->

<ns1:period xsi:type="xsd:int"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">60000</ns1:period>
</ns1:resourcePropertyElementProducers>

```

In the above, you must edit all of the argument lines.

1st argument	Specifies the absolute path location to the perf_benchmark script. It should be something like /nfs/home/user/perf_benchmark. Don't forget that this script must have executable permissions to run properly.
2nd argument	Specifies the scheduler to use. The only acceptable values here are Fork, PBS, and Condor. In theory the scheduler can be any scheduler that can be passed to WS-GRAM's globusrun-ws program.
3rd argument	Specifies the MJFS to which the job submission should be made. This must specify the MJFS to which the job submission should be made. This must be specifically a properly configured and running container. a properly configured and running container.

4th argument	Specifies the number of backend (compute) nodes that should be involved in the test that is to be run. For example, the value of 4 will run the job on 4 nodes and aggregate the results of the 4 hosts.
5th argument	Specify a valid temporary directory that the login node has file creation/removal and write access in. Some temporary files are used during the execution of this information provider and a directory must be specified where it can do this.
6th argument	Specifies the absolute path to the exec_wrapper program. It must be an absolute path and should be something like /nfs/home/user/exec_wrapper.  Important Do not forget that this script must have executable permissions to run properly.
7th argument	Specifies the test type that should be run. For this "Hello, World!" test, the value MUST be hello_world (as shown).

When this configuration block is placed properly within the resourcePropertyProviderConfiguration in the \$GLOBUS_LOCATION/etc/globus_wsrf_mds_usefulrp/rp-provider-config.xml file, the container can be restarted and when queried with a query such as this:

```
wsrf-query -s \
https://MYHOST:MYPORT/wsrf/services/DefaultIndexService \
"//*[local-name()='PerfBM-HelloWorld']"
```

You should see output that resembles the following:

```
<ns1:PerfBM-HelloWorld xmlns:ns1="http://perfbm-testing"
xmlns:exw="http://perfbm.provider/2006/execWrapper"
xmlns:pbo="http://perfbm.provider/2006/pbOutput">

  <pbo:perfBenchmarkOutputData>

    <exw:hostBenchmarkOutput>
      <exw:hostname>skynet-18</exw:hostname>
      <exw:programCommandLine>/bin/echo Hello, World!</exw:programCommandLine>
      <exw:startDate>Fri May 12 12:44:32 PDT 2006</exw:startDate>
      <exw:endDate>Fri May 12 12:44:32 PDT 2006</exw:endDate>
      <exw:testProgramOutput>
        Hello, World!
      </exw:testProgramOutput>
    </exw:hostBenchmarkOutput>

    <exw:hostBenchmarkOutput>
      <exw:hostname>skynet-19</exw:hostname>
      <exw:programCommandLine>/bin/echo Hello, World!</exw:programCommandLine>
      <exw:startDate>Fri May 12 12:44:32 PDT 2006</exw:startDate>
      <exw:endDate>Fri May 12 12:44:32 PDT 2006</exw:endDate>
      <exw:testProgramOutput>
        Hello, World!
      </exw:testProgramOutput>
    </exw:hostBenchmarkOutput>

    <exw:hostBenchmarkOutput>
```

```

<exw:hostname>skynet-94</exw:hostname>
<exw:programCommandLine>/bin/echo Hello, World!</exw:programCommandLine>
<exw:startDate>Fri May 12 12:44:32 PDT 2006</exw:startDate>
<exw:endDate>Fri May 12 12:44:32 PDT 2006</exw:endDate>
<exw:testProgramOutput>
Hello, World!
</exw:testProgramOutput>
</exw:hostBenchmarkOutput>

<exw:hostBenchmarkOutput>
<exw:hostname>skynet-93</exw:hostname>
<exw:programCommandLine>/bin/echo Hello, World!</exw:programCommandLine>
<exw:startDate>Fri May 12 12:44:32 PDT 2006</exw:startDate>
<exw:endDate>Fri May 12 12:44:32 PDT 2006</exw:endDate>
<exw:testProgramOutput>
Hello, World!
</exw:testProgramOutput>
</exw:hostBenchmarkOutput>

<pbo:perfBenchmarkErrors>
</pbo:perfBenchmarkErrors>

</pbo:perfBenchmarkOutputData>

</ns1:PerfBM-HelloWorld>

```

3.2. Configuring the Information Provider to Run the STREAM performance test

The following block of XML is the configuration that is needed to enable the Stream test in the Index Service. It uses the RPPProvider framework to create a Resource Property (RP) called PerfBM-Stream that will contain the stream output information upon successful execution.



Note

Note that this configuration block must be added to the `$GLOBUS_LOCATION/etc/globus_ws-rf_mds_usefulrp/rp-provider-config.xml` file.

```

<ns1:resourcePropertyName xsi:type="xsd:QName"
  xmlns:perfbm="http://perfbm-testing">perfbm:PerfBM-Stream</ns1:resourceProperty
  <ns1:resourcePropertyImpl
  xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.SingleValueResourcePr
  <ns1:resourcePropertyElementProducers
  xsi:type="ns1:resourcePropertyElementProducerConfig">
  <ns1:className xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">org.globus.mds.usefulrp.rpprovide

  <!-- *** SPECIFY THE SCRIPT TO RUN HERE *** -->

  <ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/perf_benchmark<

```

```

<!-- *** BEGIN SCRIPT ARGUMENTS *** -->

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">SCHEDULER</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">https://MYHOST:MYPORT/wsrf/service

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">4</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/tmp</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/exec_wrapper</n

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">stream</ns1:arguments>


<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/stream</ns1:arg

<!-- *** END ARGUMENTS *** -->

<ns1:period xsi:type="xsd:int"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">60000</ns1:period>
</ns1:resourcePropertyElementProducers>

```

In the above, you must edit all of the argument lines.

1st argument	<p>Specifies the absolute path location to the perf_benchmark script. It should be something like /nfs/home/user/perf_benchmark.</p> <p> Important</p> <p>Don't forget that this script must have executable permissions to run properly.</p>
2nd argument	<p>Specifies the scheduler to use. The only acceptable values here are Fork, PBS, and Condor. In theory the scheduler can be any scheduler that can be passed to WS-GRAM's globusrun-ws program.</p>
3rd argument	<p>Specifies the MJFS to which the job submission should be made. This must specify the MJFS to which the job submission should be made. This must be specify a properly configured and running container.</p>
4th argument	<p>Specifies the number of backend (compute) nodes that should be involved in the test that is to be run. For example, the value of 4 will run the job on 4 nodes and aggregate the results of the 4 hosts.</p>
5th argument	<p>Specify a valid temporary directory that the login node has file creation/removal and write access in. Some temporary files are used during the execution of this information provider and a directory must be specified where it can do this.</p>

6th argument	<p>Specifies the absolute path to the <code>exec_wrapper</code> program. It must be an absolute path and should be something like <code>/nfs/home/user/exec_wrapper</code>.</p> <p>! Important</p> <p>Do not forget that this script must have executable permissions to run properly.</p>
7th argument	<p>Specifies the test type that should be run. For this Stream test, the value MUST be <code>stream</code> (as shown).</p>
8th argument	<p>Specifies the absolute path location to the stream binary. It should be something like <code>/nfs/home/user/stream</code>.</p>

When this configuration block is placed properly within the `resourcePropertyProviderConfiguration` in the `$GLOBUS_LOCATION/etc/globus_wsrp/mds_usefulrp/rp-provider-config.xml` file, the container can be restarted and when queried with a query such as this:

```
wsrf-query -s https://MYHOST:MYPORT/wsrp/services/DefaultIndexService \
"//*[local-name()='PerfBM-Stream']"
```

You should see output that resembles the following:

```
<ns1:PerfBM-Stream xmlns:ns1="http://perfbm-testing"
  xmlns:exw="http://perfbm.provider/2006/execWrapper"
  xmlns:pbo="http://perfbm.provider/2006/pbOutput">
  <pbo:perfBenchmarkOutputData>
    <exw:hostBenchmarkOutput>
      <exw:hostname>skynet-5</exw:hostname>

      <exw:programCommandLine>/nfs/home/mdsdev/neillm/stream/stream_d</exw:programCo

      <exw:startDate>Mon May 15 10:03:39 PDT 2006</exw:startDate>

      <exw:endDate>Mon May 15 10:04:13 PDT 2006</exw:endDate>

      <exw:testProgramOutput>
      -----
      This system uses 8 bytes per DOUBLE PRECISION word.
      -----
      Array size = 20005000, Offset = 0
      Total memory required = 457.9 MB.
      Each test is run 10 times, but only
      the *best* time for each is used.
      -----
      Your clock granularity/precision appears to be 10000 microseconds.
      Each test below will take on the order of 579999 microseconds.
      (= 57 clock ticks)
      Increase the size of the arrays if this shows that
      you are not getting at least 20 clock ticks per test.
      -----
      WARNING -- The above is only a rough guideline.
      For best results, please be sure you know the
      precision of your system timer.
```

```
-----  
Function      Rate (MB/s)  RMS time    Min time    Max time  
Copy:        444.5556    0.7240     0.7200     0.7300  
Scale:       450.8169    0.7180     0.7100     0.7200  
Add:         558.2791    0.8650     0.8600     0.8700  
Triad:       551.8621    0.8791     0.8700     0.9100  
</exw:testProgramOutput>
```

```
</exw:hostBenchmarkOutput>
```

```
<exw:hostBenchmarkOutput>
```

```
<exw:hostname>skynet-4</exw:hostname>
```

```
<exw:programCommandLine>/nfs/home/mdsdev/neillm/stream/stream_d</exw:programCo
```

```
<exw:startDate>Mon May 15 10:03:39 PDT 2006</exw:startDate>
```

```
<exw:endDate>Mon May 15 10:04:14 PDT 2006</exw:endDate>
```

```
<exw:testProgramOutput>
```

```
-----  
This system uses 8 bytes per DOUBLE PRECISION word.  
-----
```

```
Array size = 20005000, Offset = 0  
Total memory required = 457.9 MB.  
Each test is run 10 times, but only  
the *best* time for each is used.  
-----
```

```
Your clock granularity/precision appears to be 10000 microseconds.  
Each test below will take on the order of 589999 microseconds.  
(= 58 clock ticks)
```

```
Increase the size of the arrays if this shows that  
you are not getting at least 20 clock ticks per test.  
-----
```

```
WARNING -- The above is only a rough guideline.  
For best results, please be sure you know the  
precision of your system timer.  
-----
```

```
Function      Rate (MB/s)  RMS time    Min time    Max time  
Copy:        438.4658    0.7370     0.7300     0.7400  
Scale:       438.4658    0.7300     0.7300     0.7300  
Add:         551.8621    0.8710     0.8700     0.8800  
Triad:       545.5909    0.8810     0.8800     0.8900  
</exw:testProgramOutput>
```

```
</exw:hostBenchmarkOutput>
```

```
<exw:hostBenchmarkOutput>
```

```
<exw:hostname>skynet-2</exw:hostname>
```

```
<exw:programCommandLine>/nfs/home/mdsdev/neillm/stream/stream_d</exw:programCo
```

<exw:startDate>Mon May 15 10:03:39 PDT 2006</exw:startDate>

<exw:endDate>Mon May 15 10:04:14 PDT 2006</exw:endDate>

<exw:testProgramOutput>

This system uses 8 bytes per DOUBLE PRECISION word.

Array size = 20005000, Offset = 0
Total memory required = 457.9 MB.
Each test is run 10 times, but only
the *best* time for each is used.

Your clock granularity/precision appears to be 10000 microseconds.
Each test below will take on the order of 589999 microseconds.
(= 58 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.

WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.

Function	Rate (MB/s)	RMS time	Min time	Max time
Copy:	432.5405	0.7440	0.7400	0.7500
Scale:	438.4658	0.7360	0.7300	0.7400
Add:	545.5909	0.8860	0.8800	0.8900
Triad:	539.4607	0.8960	0.8900	0.9000

</exw:testProgramOutput>

</exw:hostBenchmarkOutput>

<exw:hostBenchmarkOutput>

<exw:hostname>skynet-3</exw:hostname>

<exw:programCommandLine>/nfs/home/mdsdev/neillm/stream/stream_d</exw:programCo

<exw:startDate>Mon May 15 10:03:39 PDT 2006</exw:startDate>

<exw:endDate>Mon May 15 10:04:14 PDT 2006</exw:endDate>

<exw:testProgramOutput>

This system uses 8 bytes per DOUBLE PRECISION word.

Array size = 20005000, Offset = 0
Total memory required = 457.9 MB.
Each test is run 10 times, but only
the *best* time for each is used.

Your clock granularity/precision appears to be 10000 microseconds.
Each test below will take on the order of 589999 microseconds.
(= 58 clock ticks)

Increase the size of the arrays if this shows that you are not getting at least 20 clock ticks per test.

 WARNING -- The above is only a rough guideline.
 For best results, please be sure you know the
 precision of your system timer.

Function	Rate (MB/s)	RMS time	Min time	Max time
Copy:	432.5405	0.7440	0.7400	0.7500
Scale:	438.4658	0.7360	0.7300	0.7400
Add:	545.5909	0.8870	0.8800	0.8900
Triad:	539.4607	0.8980	0.8900	0.9000

</exw:testProgramOutput>

</exw:hostBenchmarkOutput>

<pbo:perfBenchmarkErrors>
 </pbo:perfBenchmarkErrors>

</pbo:perfBenchmarkOutputData>
 </ns1:PerfBM-Stream>

3.3. Configuring the Information Provider to Run the MPPTTEST performance test

The following block of XML is the configuration that is needed to enable the MPPTest test in the Index Service. It uses the RPPProvider framework to create a Resource Property (RP) called PerfBM-MPPTest that will contain the mpttest output information upon successful execution.

Note

Note that this configuration block must be added to the \$GLOBUS_LOCATION/etc/globus_wsrp_mds_usefulrp/rp-provider-config.xml file.

Note

Also please note that this test cannot run without a properly set MPI_LOCATION environment variable.

```
<ns1:resourcePropertyName xsi:type="xsd:QName"
  xmlns:perfbm="http://perfbm-testing">perfbm:PerfBM-MPPTest</ns1:resourceProperty>
<ns1:resourcePropertyImpl
  xsi:type="xsd:string">org.globus.mds.usefulrp.rpprovider.SingleValueResourceProperty</ns1:resourcePropertyImpl>
<ns1:resourcePropertyElementProducers
  xsi:type="ns1:resourcePropertyElementProducerConfig">
  <ns1:className xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">org.globus.mds.usefulrp.rpprovider.PerfBM-MPPTest</ns1:className>
  <!-- *** SPECIFY THE SCRIPT TO RUN HERE *** -->
  <ns1:arguments xsi:type="xsd:string"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/perf_benchmark</ns1:arguments>
```

```

<!-- *** BEGIN SCRIPT ARGUMENTS *** -->

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">SCHEDULER</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">https://MYHOST:MYPORT/wsrf/service

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">4</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/tmp</ns1:arguments>

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/exec_wrapper</n

<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">mpptest</ns1:arguments>


<ns1:arguments xsi:type="xsd:string"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">/ABSOLUTE/PATH/TO/mpptest</ns1:ar


<!-- *** END ARGUMENTS *** -->

<ns1:period xsi:type="xsd:int"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">60000</ns1:period>
</ns1:resourcePropertyElementProducers>

```

In the above, you must edit all of the argument lines.

1st argument	<p>Specifies the absolute path location to the perf_benchmark script. It should be something like /nfs/home/user/perf_benchmark.</p> <p> Important</p> <p>Don't forget that this script must have executable permissions to run properly.</p>
2nd argument	<p>Specifies the scheduler to use. The only acceptable values here are Fork, PBS, and Condor. In theory the scheduler can be any scheduler that can be passed to WS-GRAM's globusrun-ws program.</p>
3rd argument	<p>Specifies the MJFS to which the job submission should be made. This must be specify a properly configured and running container..</p>
4th argument	<p>Specifies the number of backend (compute) nodes that should be involved in the test that is to be run. For example, the value of 4 will run the job on 4 nodes and aggregate the results of the 4 hosts.</p>
5th argument	<p>Specify a valid temporary directory that the login node has file creation/removal and write access in. Some temporary files are used during the execution of this information provider and a directory must be specified where it can do this.</p>

6th argument	<p>Specifies the absolute path to the <code>exec_wrapper</code> program. It must be an absolute path and should be something like <code>/nfs/home/user/exec_wrapper</code>.</p> <p> Important</p> <p>Do not forget that this script must have executable permissions to run properly.</p>
7th argument	Specifies the test type that should be run. For this MPPTest test, the value MUST be <code>mpptest</code> (as shown).
8th argument	Specifies the absolute path location to the stream binary. It should be something like <code>/nfs/home/user/mpptest</code> .

When this configuration block is placed properly within the `resourcePropertyProviderConfiguration` in the `$GLOBUS_LOCATION/etc/globus_wsrp/mds_usefulrp/rp-provider-config.xml` file, the container can be restarted and when queried with a query such as this:

```
wsrf-query -s https://MYHOST:MYPORT/wsrp/services/DefaultIndexService \
"//*[local-name()='PerfBM-MPPTest']"
```

You should see output that resembles the following:

```
<ns1:PerfBM-MPPTest xmlns:ns1="http://perfbm-testing"
  xmlns:exw="http://perfbm.provider/2006/execWrapper"
  xmlns:pbo="http://perfbm.provider/2006/pbOutput">
  <pbo:perfBenchmarkOutputData>
  <exw:hostBenchmarkOutput>
  <exw:hostname>skynet-2</exw:hostname>

  <exw:programCommandLine>/nfs/software/mpich/1.2.7/bin/mpirun -np 4
  /nfs/home/mdsdev/neillm/bin/mpptest</exw:programCommandLine>

  <exw:startDate>Wed May 17 11:52:35 PDT 2006</exw:startDate>

  <exw:endDate>Wed May 17 11:52:52 PDT 2006</exw:endDate>

  <exw:testProgramOutput>
  set default
  set font variable
  set curve window y 0.15 0.90
  set order d d d x y d
  title left 'time (us)', bottom 'Size (bytes)',
  top 'Comm Perf for MPI (skynet-2.isi.edu)',
  'type = blocking'

  #p0    p1    dist    len    ave time (us)    rate
  0      3      3       0      70.340000        0.00
  0      3      3      32      72.750000       439.863e+3
  0      3      3      64     106.750000       599.532e+3
  0      3      3      96     123.070000       780.044e+3
  0      3      3     128     124.070000       1.032e+6
  0      3      3     160     124.170000       1.289e+6
  0      3      3     192     124.270000       1.545e+6
```

0	3	3	224	124.300000	1.802e+6
0	3	3	256	124.490000	2.056e+6
0	3	3	288	124.510000	2.313e+6
0	3	3	320	124.450000	2.571e+6
0	3	3	352	124.750000	2.822e+6
0	3	3	384	124.810000	3.077e+6
0	3	3	416	125.150000	3.324e+6
0	3	3	448	125.390000	3.573e+6
0	3	3	480	126.970000	3.780e+6
0	3	3	512	128.340000	3.989e+6
0	3	3	544	133.480000	4.076e+6
0	3	3	576	136.000000	4.235e+6
0	3	3	608	137.880000	4.410e+6
0	3	3	640	163.280000	3.920e+6
0	3	3	672	185.420000	3.624e+6
0	3	3	704	186.560000	3.774e+6
0	3	3	736	186.730000	3.942e+6
0	3	3	768	186.740000	4.113e+6
0	3	3	800	186.690000	4.285e+6
0	3	3	832	186.820000	4.453e+6
0	3	3	864	186.900000	4.623e+6
0	3	3	896	187.000000	4.791e+6
0	3	3	928	187.110000	4.960e+6
0	3	3	960	187.200000	5.128e+6
0	3	3	992	187.250000	5.298e+6
0	3	3	1024	187.230000	5.469e+6

```
plot square
```

```
join
```

```
wait
```

```
new page
```

```
</exw:testProgramOutput>
```

```
</exw:hostBenchmarkOutput>
```

```
<exw:hostBenchmarkOutput>
```

```
<exw:hostname>skynet-3</exw:hostname>
```

```
<exw:programCommandLine>/nfs/software/mpich/1.2.7/bin/mpirun -np 4  
/nfs/home/mdsdev/neillm/bin/mpptest</exw:programCommandLine>
```

```
<exw:startDate>Wed May 17 11:52:35 PDT 2006</exw:startDate>
```

```
<exw:endDate>Wed May 17 11:52:52 PDT 2006</exw:endDate>
```

```
<exw:testProgramOutput>
```

```
set default
```

```
set font variable
```

```
set curve window y 0.15 0.90
```

```
set order d d d x y d
```

```
title left 'time (us)', bottom 'Size (bytes)',
```

```
top 'Comm Perf for MPI (skynet-3.isi.edu)',
```

```
'type = blocking'
```

#p0	p1	dist	len	ave time (us)	rate
0	3	3	0	94.780000	0.00
0	3	3	32	96.070000	333.090e+3
0	3	3	64	106.750000	599.532e+3
0	3	3	96	124.140000	773.320e+3
0	3	3	128	124.200000	1.031e+6
0	3	3	160	124.180000	1.288e+6
0	3	3	192	124.220000	1.546e+6
0	3	3	224	124.260000	1.803e+6
0	3	3	256	124.440000	2.057e+6
0	3	3	288	124.510000	2.313e+6
0	3	3	320	124.540000	2.569e+6
0	3	3	352	124.580000	2.825e+6
0	3	3	384	124.660000	3.080e+6
0	3	3	416	125.700000	3.309e+6
0	3	3	448	125.610000	3.567e+6
0	3	3	480	128.180000	3.745e+6
0	3	3	512	131.910000	3.881e+6
0	3	3	544	139.550000	3.898e+6
0	3	3	576	150.530000	3.826e+6
0	3	3	608	152.130000	3.997e+6
0	3	3	640	163.000000	3.926e+6
0	3	3	672	185.490000	3.623e+6
0	3	3	704	186.480000	3.775e+6
0	3	3	736	186.630000	3.944e+6
0	3	3	768	186.660000	4.114e+6
0	3	3	800	186.710000	4.285e+6
0	3	3	832	186.770000	4.455e+6
0	3	3	864	186.930000	4.622e+6
0	3	3	896	187.020000	4.791e+6
0	3	3	928	187.040000	4.962e+6
0	3	3	960	188.210000	5.101e+6
0	3	3	992	187.980000	5.277e+6
0	3	3	1024	187.180000	5.471e+6

plot square

join

wait

new page

</exw:testProgramOutput>

</exw:hostBenchmarkOutput>

<exw:hostBenchmarkOutput>

<exw:hostname>skynet-4</exw:hostname>

<exw:programCommandLine>/nfs/software/mpich/1.2.7/bin/mpirun -np 4
/nfs/home/mdsdev/neillm/bin/mpptest</exw:programCommandLine>

<exw:startDate>Wed May 17 11:52:35 PDT 2006</exw:startDate>

<exw:endDate>Wed May 17 11:52:52 PDT 2006</exw:endDate>

<exw:testProgramOutput>

```
set default
set font variable
set curve window y 0.15 0.90
set order d d d x y d
title left 'time (us)', bottom 'Size (bytes)',
top 'Comm Perf for MPI (skynet-4.isi.edu)',
'type = blocking'
```

#p0	p1	dist	len	ave time (us)	rate
0	3	3	0	77.630000	0.00
0	3	3	32	95.880000	333.751e+3
0	3	3	64	105.720000	605.373e+3
0	3	3	96	121.670000	789.019e+3
0	3	3	128	124.090000	1.032e+6
0	3	3	160	124.240000	1.288e+6
0	3	3	192	124.280000	1.545e+6
0	3	3	224	124.250000	1.803e+6
0	3	3	256	124.510000	2.056e+6
0	3	3	288	124.490000	2.313e+6
0	3	3	320	124.710000	2.566e+6
0	3	3	352	124.540000	2.826e+6
0	3	3	384	124.820000	3.076e+6
0	3	3	416	124.840000	3.332e+6
0	3	3	448	125.370000	3.573e+6
0	3	3	480	126.530000	3.794e+6
0	3	3	512	129.260000	3.961e+6
0	3	3	544	136.720000	3.979e+6
0	3	3	576	150.300000	3.832e+6
0	3	3	608	151.210000	4.021e+6
0	3	3	640	160.530000	3.987e+6
0	3	3	672	186.560000	3.602e+6
0	3	3	704	186.530000	3.774e+6
0	3	3	736	186.680000	3.943e+6
0	3	3	768	186.740000	4.113e+6
0	3	3	800	186.880000	4.281e+6
0	3	3	832	187.010000	4.449e+6
0	3	3	864	187.180000	4.616e+6
0	3	3	896	187.000000	4.791e+6
0	3	3	928	187.190000	4.958e+6
0	3	3	960	187.210000	5.128e+6
0	3	3	992	187.200000	5.299e+6
0	3	3	1024	187.200000	5.470e+6

```
plot square
```

```
join
```

```
wait
```

```
new page
```

```
</exw:testProgramOutput>
```

```
</exw:hostBenchmarkOutput>
```

```
<exw:hostBenchmarkOutput>
```

```
<exw:hostname>skynet-5</exw:hostname>
```

```
<exw:programCommandLine>/nfs/software/mpich/1.2.7/bin/mpirun -np 4
/nfs/home/mdsdev/neillm/bin/mpptest</exw:programCommandLine>
```

```
<exw:startDate>Wed May 17 11:52:35 PDT 2006</exw:startDate>
```

```
<exw:endDate>Wed May 17 11:52:52 PDT 2006</exw:endDate>
```

```
<exw:testProgramOutput>
```

```
set default
```

```
set font variable
```

```
set curve window y 0.15 0.90
```

```
set order d d d x y d
```

```
title left 'time (us)', bottom 'Size (bytes)',
```

```
top 'Comm Perf for MPI (skynet-5.isi.edu)',
```

```
'type = blocking'
```

#p0	p1	dist	len	ave time (us)	rate
0	3	3	0	93.560000	0.00
0	3	3	32	94.490000	338.660e+3
0	3	3	64	106.770000	599.419e+3
0	3	3	96	124.160000	773.196e+3
0	3	3	128	123.250000	1.039e+6
0	3	3	160	124.260000	1.288e+6
0	3	3	192	124.250000	1.545e+6
0	3	3	224	124.310000	1.802e+6
0	3	3	256	124.410000	2.058e+6
0	3	3	288	124.470000	2.314e+6
0	3	3	320	124.520000	2.570e+6
0	3	3	352	124.730000	2.822e+6
0	3	3	384	125.580000	3.058e+6
0	3	3	416	125.630000	3.311e+6
0	3	3	448	125.200000	3.578e+6
0	3	3	480	127.930000	3.752e+6
0	3	3	512	127.280000	4.023e+6
0	3	3	544	129.450000	4.202e+6
0	3	3	576	143.250000	4.021e+6
0	3	3	608	155.510000	3.910e+6
0	3	3	640	164.380000	3.893e+6
0	3	3	672	185.450000	3.624e+6
0	3	3	704	186.550000	3.774e+6
0	3	3	736	186.610000	3.944e+6
0	3	3	768	186.680000	4.114e+6
0	3	3	800	186.780000	4.283e+6
0	3	3	832	186.770000	4.455e+6
0	3	3	864	186.910000	4.623e+6
0	3	3	896	186.990000	4.792e+6
0	3	3	928	187.000000	4.963e+6
0	3	3	960	187.100000	5.131e+6
0	3	3	992	187.420000	5.293e+6
0	3	3	1024	187.270000	5.468e+6

```
plot square
```

```
join
```

```
wait
```

```
new page
```

```
</exw:testProgramOutput>

</exw:hostBenchmarkOutput>

<pbo:perfBenchmarkErrors>
</pbo:perfBenchmarkErrors>

</pbo:perfBenchmarkOutputData>
</nsl:PerfBM-MPPTest>
```

3.3.1. Troubleshooting the MPPTTEST performance test

If you are seeing the following error after the container has been started, the most likely cause is that you have not properly set the `MPI_LOCATION` environment variable. Please set this to a suitable MPI location (such as `/nfs/software/mpich-1.2.7`) and restart the container.

```
2006-05-17 11:46:38,371 INFO impl.DefaultIndexService
[ServiceThread-12,processConfigFile:107] Reading default registration
configuration from file:
/scratch/mdsdev-neillm/gt4.0.2-plus-cvs/etc/globus_wsrf_mds_index/hierarch
2006-05-17 11:46:38,702 ERROR rprovider.ResourcePropertyProviderTask
[Thread-16,timerExpired:159] Unhandled exception during execution of
org.globus.mds.usefulrp.rprovider.producers.ExternalProcessElementProduce
java.lang.Exception: Exception while parsing child process stdout into
valid XML document: org.xml.sax.SAXException: Fatal Error: URI=null
Line=-1: Premature end of file.
```

3.4. Configuring the Information Provider to Run the MPPTTEST Logscale performance tests

The configuration for this test is exactly the same as the configuration for the [MPPTest](#) above, except for the RP name and 7th argument:

- Change the RP name from `PerfBM-MPPTest` to `PerfBM-MPPTest-LogScale`.
- Change the 7th argument of the `$GLOBUS_LOCATION/etc/globus_wsrf_mds_usefulrp/rp-provider-config.xml` file to be `mpptest-logscale` instead of `mpptest`.

3.5. Configuring the Information Provider to Run the MPPTTEST Bisect Logscale performance tests

The configuration for this test is exactly the same as the configuration for the [MPPTest](#) above, except for the RP name and 7th argument:

- Change the RP name from `PerfBM-MPPTest` to `PerfBM-MPPTest-BiSect-LogScale`.
- Change the 7th argument of the `$GLOBUS_LOCATION/etc/globus_wsrf_mds_usefulrp/rp-provider-config.xml` file to be `mpptest-bisect-logscale` instead of `mpptest`.

3.6. Configuring the Information Provider to Run the MPIBENCH/LLCBENCH performance tests

Information on this particular test is to be available at a future date.

4. Resource Properties

TODO: the resource properties the provider creates/collects/advertises/publishes

4.1. Namespace URI

TODO: Include the Namespace URI, or provide a link to the Java API Documentation, which is required to construct the QName for the subscription to the Topic.

5. Schema

TODO: link to schema files

6. Security Considerations

General security considerations associated with the container and all MDS services apply. See: [Aggregator Framework](#).

Additionally the user should consider that publishing system specific information as is the intention of this provider can have security ramifications.

6.1. WS MDS Aggregator Services (Index Service and Trigger Service) Security Considerations

By default, the *aggregator sources* do not use authentication credentials -- they retrieve information using anonymous SSL authentication or no authentication at all, and thus retrieve only publicly-available information. If a user or administrator changes that configuration so that a service's aggregator source uses credentials to acquire non-privileged data, then that user or administrator must configure the service's aggregator sink to limit access to authorized users.

7. Testing

TODO: add a simple test for this info provider

8. Troubleshooting

TODO: describe common issues users may experience with this info provider

Glossary

A

aggregator source

A Java class that implements an interface (defined as part of the Aggregator Framework) to collect XML-formatted data. WS MDS contains three aggregator sources: the query aggregator source, the subscription aggregator source, and the execution aggregator source.