高エネルギー加速器研究機構　計算科学センター

Globus Toolkit インストール作業ログ

高瀬　亘

2011/11/16

目次

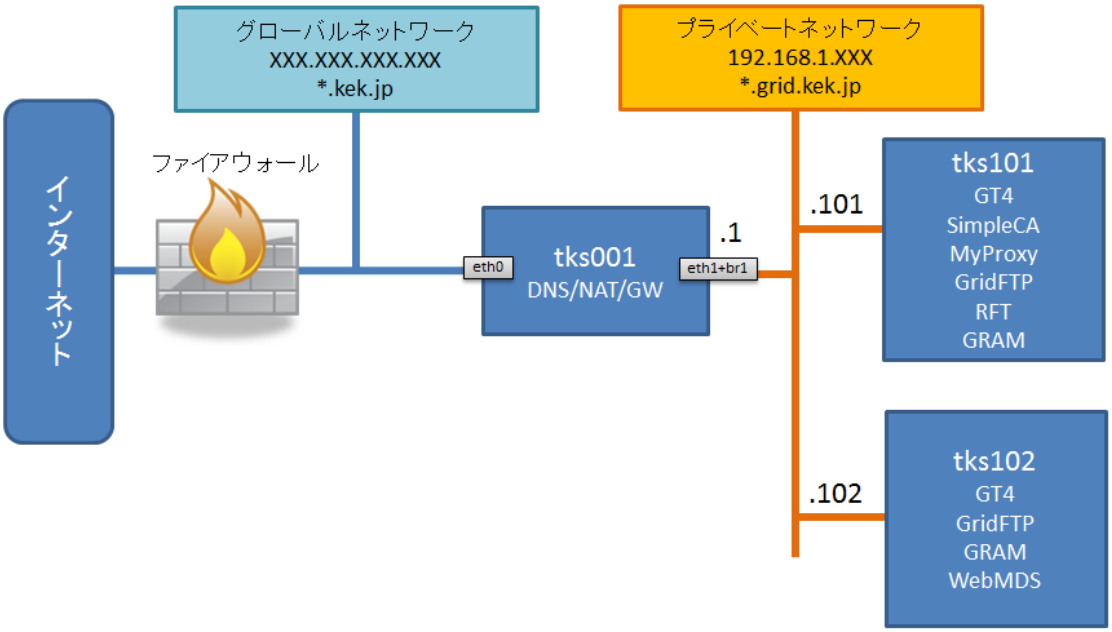
1. はじめに 2
2. ネットワーク構成 3
3. 事前準備 4
3.1. ユーザの追加 4
3.1.1. グループ globus の追加 4
3.1.2. ユーザ hoge の追加 4
3.1.3. ユーザ globus の追加 4
3.2. JDK のインストール 4
3.3. Ant のインストール 4
3.4. GCC の確認 5
3.5. GNU tar の確認 5
3.6. GNU sed の確認 5
3.7. zlib の確認 5
3.8. GNU make の確認 5
3.9. sudo の確認 6
3.10. openssl 及び openssl-devel の確認 6
3.11. postgreSQL のインストール 6
3.12. iODBC のインストール 7
3.13. psqlodbc ドライバのインストール 7
4. Globus Toolkit4 のインストール 8
4.1. Globus Toolkit4 のインストール 8
4.2. SimpleCA の設定 8
4.3. ホスト証明書の発行 8
4.4. ホスト証明書の配置 9
4.5. SimpleCA 証明書の配置 9
4.6. ユーザ証明書及び KEK CA 証明書の配置 10
4.7. grid-mapfile の設定 10
4.7.1. grid-mapfile の設定(tks101) 10
4.7.2. grid-mapfile の設定(tks102) 11
4.8. MyProxy サーバの作成 11
4.9. GridFTP の設定 11
4.10. Web Service コンテナの設定 12
4.10.1. tks101 上からの確認 12
4.10.2. tks102 上からの確認 12
4.11. RFT の設定 13
4.12. GRAM4 の設定 13
4.12.1. GRAM4 の設定 13
4.12.2. GRAM4 のテスト 14
4.13. MDS の設定 15
4.14. WebMDS の設定 15

1. はじめに

本書は、過去に筆者がGlobus Toolkitをインストールした時の作業ログに修正をほどこし、公開可能にしたものです。Globus Toolkitのインストールに関する情報を網羅的に記載しています。

2. ネットワーク構成

- 本書では、1 台のホストマシン(tks001)上に VM(Virtual Machine)を 2 台(tks101, tks102)立ち上げ、VM 上に Globus Toolkit をインストールする
- OS は、ホストマシンおよびゲストマシンともに CentOS 5.6(64bit)を使用している



3. 事前準備

3.1. ユーザの追加

・ tks101, tks102 上, ユーザ root で行う

3.1.1. グループ globus の追加

```
# groupadd -g 2000 globus
```

3.1.2. ユーザ hoge の追加

```
# useradd -c 'HOGE Hogera' -d /home/hoge -m -g globus -u 1000 hoge
# id hoge
# passwd hoge
```

3.1.3. ユーザ globus の追加

```
# useradd -g globus -u 2000 globus
# passwd globus
```

3.2. JDK のインストール

・ tks101, tks102 上, ユーザ root で行う

```
# ls /tmp/jdk*
  /tmp/jdk-6u24-linux-x64.bin
# mkdir /usr/local/java/
# cd /usr/local/java/
# cp /tmp/jdk-6u24-linux-x64.bin .
# chmod u+x jdk-6u24^linux-x64.bin
# vi /etc/profile
  export JAVA_HOME=/usr/local/java/jdk1.6.0_24  ←追加
  export PATH=$PATH:$JAVA_HOME/bin  ←追加
# source /etc/profile
# java -version
  java version "1.6.0_24"
  Java(TM) SE Runtime Environment (build 1.6.0_24-b07)
  Java HotSpot(TM) 64-Bit Server VM (build 19.1-b02, mixed mode)
```

3.3. Ant のインストール

・ tks101, tks102 上, ユーザ root で行う

```
# mkdir /usr/local/ant
# cd /usr/local/ant
# wget ¥
http://ftp.jaist.ac.jp/pub/apache//ant/binaries/apache-ant-1.8.2-bin.tar.gz
```

```
# tar zxvf apache-ant-1.8.2-bin.tar.gz
# vi /etc/profile
  export ANT_HOME=/usr/local/ant/apache-ant-1.8.2  ←追加
  export PATH=$PATH:$JAVA_HOME/bin:$ANT_HOME/bin  ←修正
# source /etc/profile
# ant -version
 Apache Ant(TM) version 1.8.2 compiled on December 20 2010
```

3.4. GCC の確認

・ tks101, tks102 上

```
# gcc -dumpversion
 4.1.2
```

3.5. GNU tar の確認

・ tks101, tks102 上

```
# tar --version
 tar (GNU tar) 1.15.1
```

3.6. GNU sed の確認

・ tks101, tks102 上

```
# sed --version
 GNU sed version 4.1.5
 -- snipped --
```

3.7. zlib の確認

・ tks101, tks102 上

```
# less /usr/include/zlib.h
 -- snipped --
   version 1.2.3, July 18th, 2005
 -- snipped --
```

3.8. GNU make の確認

・ tks101, tks102 上

```
# make --version
 GNU Make 3.81
 -- snipped --
```

3.9.　sudo の確認

・　tks101, tks102 上

```
# sudo -V
  Sudo version 1.7.2p1
  -- snipped --
```

3.10. openssl 及び openssl-devel の確認

・　tks101, tks102 上, ユーザ root で行う

```
# yum list installed | grep openssl
  openssl.i686                    0.9.8e-12.el5_4.6        installed
  openssl.x86_64                  0.9.8e-12.el5_4.6        installed
  openssl-devel.i386              0.9.8e-12.el5_4.6        installed
  openssl-devel.x86_64            0.9.8e-12.el5_4.6        installed
  xmlsec1-openssl.i386            1.2.9-8.1.1              installed
  xmlsec1-openssl.x86_64          1.2.9-8.1.1              installed
  xmlsec1-openssl-devel.i386      1.2.9-8.1.1              installed
  xmlsec1-openssl-devel.x86_64    1.2.9-8.1.1              installed
```

3.11. postgreSQL のインストール

・　tks101 上, ユーザ root で行う

```
# yum remove postgresql
# useradd postgres
# passwd postgres
# mkdir /usr/local/pgsql
# chown postgres:postgres /usr/local/pgsql/
# cd /usr/local/src/
# wget ¥
http://wwwmaster.postgresql.org/redir/323/f/source/¥
v9.0.4/postgresql-9.0.4.tar.gz
# tar zxvf postgresql-9.0.4.tar.gz
# chown postgres:postgres postgresql-9.0.4
# su - postgres
$ cd /usr/local/src/postgresql-9.0.4
$ ./configure
$ make
$ make install
$ vi ~/.bash_profile
  POSTGRE_PATH=/usr/local/pgsql
  PGDATA=$POSTGRE_PATH/data
$ . ~/.bash_profile
```

3.12. iODBC のインストール

・ tks101 上, ユーザ root で行う

```
# cd /usr/local/src/
# wget http://www.iodbc.org/downloads/iODBC/libiodbc-3.52.7.tar.gz
# tar zxvf libiodbc-3.52.7.tar.gz
# chown postgres:postgres libiodbc-3.52.7
# su - postgres
$ cd /usr/local/src/libiodbc-3.52.7
$ ./configure
$ make
$ logout
# cd /usr/local/src/libiodbc-3.52.7
# make instal
# vi /etc/ld.so.conf
  /usr/local/lib  ←追加
  /usr/local/pgsql/lib  ←追加
# ldconfig
# ldconfig -v | grep libiodbc.so.2
```

3.13. psqlodbc ドライバのインストール

・ tks101 上, ユーザ root で行う

```
# cd /usr/local/src/
# wget ¥
http://wwwmaster.postgresql.org/redir/323/f/odbc/versions/src/¥
psqlodbc-09.00.0300.tar.gz
# tar zxvf psqlodbc-09.00.0300.tar.gz
# chown postgres:postgres psqlodbc-09.00.0300
# su - postgres
$ cd /usr/local/src/psqlodbc-09.00.0300/
$ ./configure --with-iodbc --with-libpq=/usr/local/pgsql/
$ make
$ logout
# cd /usr/local/src/psqlodbc-09.00.0300/
# make install
```

4. Globus Toolkit4 のインストール

4.1. Globus Toolkit4 のインストール

・ tks101, tks102 上, ユーザ root で行う

```
# ls /tmp/gt4*
  /tmp/gt4.2.1-all-source-installer.tar.bz2
# chown globus:globus /tmp/gt4.2.1-all-source-installer.tar.bz2
# mkdir /usr/local/globus-4.2.1
# chown globus:globus /usr/local/globus-4.2.1/
# su - globus
$ cd /tmp/
$ tar xvf gt4.2.1-all-source-installer.tar.bz2
$ cd gt4.2.1-all-source-installer
$ export GLOBUS_LOCATION=/usr/local/globus-4.2.1
$ ./configure --prefix=$GLOBUS_LOCATION
$ make
$ make install
$ logout
# vi /etc/profile.d/set_globus_env.sh
  export GLOBUS_LOCATION=/usr/local/globus-4.2.1  ←追加
. $GLOBUS_LOCATION/etc/globus-user-env.sh
```

4.2. SimpleCA の設定

・ tks101 上, ユーザ root で行う

```
# su - globus
$ $GLOBUS_LOCATION/setup/globus/setup-simple-ca
  CA subject : cn=Globus Simple CA, ou=simpleCA-tks101.grid.kek.jp, ou=GlobusTest,
  o=Grid
  Email of the CA : globus@localhost
  CA certificate of expiration date :
  PEM pass phrase:
$ logout
# $GLOBUS_LOCATION/setup/globus_simple_ca_1ebe31a8_setup/setup-gsi -default
# scp /etc tks102:/tmp/
```

4.3. ホスト証明書の発行

・ tks101 上, ユーザ root で行う

```
# grid-cert-request -host 'tks101.grid.kek.jp' -dir /tmp/cert/host/tks101
# grid-cert-request -host 'tks102.grid.kek.jp' -dir /tmp/cert/host/tks102
# chown globus:globus /tmp/cert/host/tks101
```

```
# chown globus:globus /tmp/cert/host/tks102
$ su - globus
$ cd
$ grid-ca-sign -in /tmp/cert/host/tks101/hostcert_request.pem ¥
-out /tmp/cert/host/tks101/hostsigned.pem
$ grid-ca-sign -in /tmp/cert/host/tks102/hostcert_request.pem ¥
-out /tmp/cert/host/tks102/hostsigned.pem
$ logout
# cp /tmp/cert/host/tks101/hostsigned.pem /etc/grid-security/hostcert.pem
# cp /tmp/cert/host/tks101/hostkey.pem /etc/grid-security/hostkey.pem
# cd /etc/grid-security/
# chmod 644 hostcert.pem
# chown root:root host*
# chmod 400 hostkey.pem
# rm -f hostcert_request.pem
# cp hostcert.pem containercert.pem
# cp hostkey.pem containerkey.pem
# chown globus:globus container*
```

4.4. ホスト証明書の配置

・ tks102 上, ユーザ root で行う

```
# mkdir /etc/grid-security
# scp -r tks101:/tmp/cert/host/tks102/* /etc/grid-security/
# cd /etc/grid-security/
# mv hostsigned.pem hostcert.pem
# chmod 644 hostcert.pem
# chown root:root host*
# chmod 400 hostkey.pem
# rm -f hostcert_request.pem
# cp hostcert.pem containercert.pem
# cp hostkey.pem containerkey.pem
# chown globus:globus container*
```

4.5. SimpleCA 証明書の配置

・ tks102 上, ユーザ root で行う

```
# mkdir /etc/grid-security/certificates
# scp tks101:/etc/grid-security/certificates/1ebe31a8.* ¥
/etc/grid-security/certificates/
```

4.6.　ユーザ証明書及びKEK CA証明書の配置

・　tks101, tks102上, ユーザrootで行う

※/tmp/cert/にKEK CAより発行されたユーザ証明書とKEK CAの証明書を置いておく

```
# ls -R /tmp/cert/
 /tmp/cert/:
 ca  user

 /tmp/cert/cacert:
 kek

 /tmp/cert/ca/kek:
 617ff41b.0  617ff41b.signing_policy

 /tmp/cert/user:
 hoge

 /tmp/cert/user/hoge:
 usercert.pem  userkey.pem
# cp /tmp/cert/ca/kek/617ff41b.* /etc/grid-security/certificates/
# chown root:root /etc/grid-security/certificates/*
# chmod 644 /etc/grid-security/certificates/*
# cat /etc/grid-security/certificates/617ff41b.signing_policy
  access_id_CA      X509 ¥
  '/C=JP/O=KEK/OU=CRC/CN=KEK GRID Certificate Authority'
  pos_rights        globus        CA:sign
  cond_subjects     globus ¥
  '"/C=JP/O=KEK/OU=CRC/CN=KEK GRID Certificate Authority" "/C=JP/O=KEK/OU=CRC/*"'
# mkdir ~hoge/.globus
# cp /tmp/cert/user/hoge/user* ~hoge/.globus/
# chown -R hoge:globus ~hoge/.globus/
# chmod 644 ~hoge/.globus/usercert.pem
# chmod 400 ~hoge/.globus/userkey.pem
# su - hoge
$ grid-proxy-init -debug -verify
```

4.7.　grid-mapfileの設定

・　ユーザrootで行う

4.7.1. grid-mapfileの設定(tks101)

```
# vi /etc/grid-security/grid-mapfile
"/C=JP/O=KEK/OU=CRC/CN=HOGE Hogera" hoge
```

```
"/O=Grid/OU=GlobusTest/OU=simpleCA-tks101.grid.kek.jp/CN=host/¥
tks101.grid.kek.jp" globus
```

4.7.2. grid-mapfile の設定(tks102)

```
# vi /etc/grid-security/grid-mapfile
"/C=JP/O=KEK/OU=CRC/CN=HOGE Hogera" hoge
"/O=Grid/OU=GlobusTest/OU=simpleCA-tks101.grid.kek.jp/CN=host/¥
tks102.grid.kek.jp" globus
```

4.8. MyProxy サーバの作成

・ tks101 上, ユーザ root で行う

```
# cp $GLOBUS_LOCATION/share/myproxy/myproxy-server.config /etc
# vi /etc/myproxy-server.config
  accepted_credentials       "*"  ←コメント外す
  authorized_retrievers      "*"  ←コメント外す
  default_retrievers         "*"  ←コメント外す
  authorized_renewers        "*"  ←コメント外す
  default_renewers           "none"  ←コメント外す
# vi /etc/services
  myproxy-server  7512/tcp  ←追加
# cp /usr/local/globus-4.2.1/share/myproxy/etc.init.d.myproxy ¥
/etc/rc.d/init.d/myproxy
# chmod a+x /etc/rc.d/init.d/myproxy
# chkconfig --add myproxy
# chkconfig --list myproxy
# vi /etc/rc.d/init.d/myproxy
  GLOBUS_LOCATION="/usr/local/globus-4.2.1"  ←修正
# /etc/init.d/myproxy start
# netstat -an | grep 7512
  tcp        0      0 0.0.0.0:7512            0.0.0.0:*                 LISTEN
# su - hoge
$ myproxy-init -a -s tks101.grid.kek.jp
$ myproxy-logon -s tks101.grid.kek.jp
```

4.9. GridFTP の設定

・ tks101, tks102 上, ユーザ root で行う

```
# vi /etc/xinetd.d/gridftp
  service gsiftp
  {
    instances               = 100
```

```
    socket_type             = stream
    wait                    = no
    user                    = root
    env                     += GLOBUS_LOCATION=/usr/local/globus-4.2.1
    env                     += LD_LIBRARY_PATH=/usr/local/globus-4.2.1/lib 2
    env                     += GLOBUS_TCP_PORT_RANGE=20000,25000
    server                  = /usr/local/globus-4.2.1/sbin/globus-gridftp-server
    server_args             = -i
    log_on_success          += DURATION
    disable                 = no
  }
# vi /etc/services
  gsiftp          2811/tcp  ←追加
# /etc/init.d/xinetd reload
# netstat -an | grep 2811
  tcp        0      0 0.0.0.0:2811            0.0.0.0:*                 LISTEN
# su - hoge
$ cd
$ date > hoge
$ grid-proxy-init
$ globus-url-copy gsiftp://tks101.grid.kek.jp/home/hoge/hoge ¥
gsiftp://tks102.grid.kek.jp/tmp/hoge
$ ssh tks102
$ cat /tmp/hoge
Mon Jul 11 17:10:08 JST 2011
```

4.10. Web Service コンテナの設定

・ tks101, tks102 上, ユーザ root で行う

```
# cp $GLOBUS_LOCATION/etc/init.d/globus-ws-java-container /etc/init.d/
# su - globus
$ /etc/init.d/globus-ws-java-container start  ←一度電源を落とすと手動で起動しないといけない？
```

4.10.1. tks101 上からの確認

```
$ globus-check-remote-environment -s https://tks101.grid.kek.jp:8443
$ globus-check-remote-environment -s https://tks102.grid.kek.jp:8443
```

4.10.2. tks102 上からの確認

```
$ globus-check-remote-environment -s https://tks102.grid.kek.jp:8443
$ globus-check-remote-environment -s https://tks101.grid.kek.jp:8443
```

4.11. RFT の設定

・ tks101 上，ユーザ root で行う

```
$ grid-proxy-init
 Your identity: /C=JP/O=KEK/OU=CRC/CN=HOGE Hogera
 Enter GRID pass phrase for this identity:
 Creating proxy ..................................... Done
 Your proxy is valid until: Tue Jul 12 04:23:29 2011
$ cat transfer
 gsiftp://tks101.grid.kek.jp/etc/group gsiftp://tks101.grid.kek.jp/tmp/asdf
$ globus-crft -ez -f transfer
 Communicating with delegation service.
 Creating the RFT service.
 Starting the RFT service.
 Waiting for the RFT transfers to complete.
 Transfered 1 of 1                        | Status: Done
$ diff /etc/group /tmp/asdf
$ less /tmp/asdf
```

4.12. GRAM4 の設定

4.12.1. GRAM4 の設定

・ tks101，tks102 上，ユーザ root で行う

```
# visudo
 # Defaults    requiretty  ←コメントアウト
 globus ALL=(hoge) ¥
 NOPASSWD: /usr/local/globus-4.2.1/libexec/globus-gridmap-and-execute ¥
 -g /etc/grid-security/grid-mapfile ¥
 /usr/local/globus-4.2.1/libexec/globus-job-manager-script.pl *  ←追加
 globus ALL=(hoge) ¥
 NOPASSWD: /usr/local/globus-4.2.1/libexec/globus-gridmap-and-execute ¥
 -g /etc/grid-security/grid-mapfile ¥
 /usr/local/globus-4.2.1/libexec/globus-gram-local-proxy-tool *  ←追加
$ su - hoge
$ globusrun-ws -submit -c /bin/true
 Submitting job...Done.
 Job ID: uuid:e51f4f54-ab92-11e0-942f-545200001bb3
 Termination time: 07/11/3011 07:53 GMT
 Current job state: Active
 Current job state: CleanUp
 Current job state: Done
 Destroying job...Done.
$ echo $?
```

```
 0
$ globusrun-ws -submit -c /bin/false
 Submitting job...Done.
 Job ID: uuid:ebcc7890-ab92-11e0-8633-545200001bb3
 Termination time: 07/11/3011 07:53 GMT
 Current job state: Active
 Current job state: CleanUp
 Current job state: Done
 Destroying job...Done.
$ echo $?
 1
```

4.12.2. GRAM4 のテスト

- tks101, tks102 上, ユーザ hoge で行う
- 下記は tks101 上の場合（tks102 上では、tks102 の部分を tks101 に変える）

```
$ vi stage.xml
 <job>
  <executable>my_hostname</executable>
  <directory>${GLOBUS_USER_HOME}</directory>
  <stdout>${GLOBUS_USER_HOME}/stdout</stdout>
  <stderr>${GLOBUS_USER_HOME}/stderr</stderr>
  <fileStageIn>
   <transfer>
    <sourceUrl>gsiftp://tks101.grid.kek.jp:2811/bin/hostname</sourceUrl>
    <destinationUrl>file:///${GLOBUS_USER_HOME}/my_hostname</destinationUrl>
   </transfer>
  </fileStageIn>
  <fileStageOut>
   <transfer>
    <sourceUrl>file:///${GLOBUS_USER_HOME}/stdout</sourceUrl>
    <destinationUrl>gsiftp://tks101.grid.kek.jp:2811/tmp/out</destinationUrl>
   </transfer>
  </fileStageOut>
  <fileCleanUp>
   <deletion>
    <file>file:///${GLOBUS_USER_HOME}/my_hostname</file>
   </deletion>
  </fileCleanUp>
 </job>
$ grid-proxy-init
$ GLOBUS_USER_HOME=/home/hoge
$ globusrun-ws -submit -S -f stage.xml
```

```
$ cat out
  tks101.grid.kek.jp
$ ssh tks102
$ cat /tmp/out
  tks101.grid.kek.jp
```

4.13. MDS の設定

・ tks102 上，ユーザ root で行う

```
# su - globus
$ vi /usr/local/globus-4.2.1/etc/globus_wsrf_mds_index/hierarchy.xml
```

※下記を追加

```
  <upstream>
     https://tks101.grid.kek.jp:8443/wsrf/services/DefaultIndexService
  </upstream>
$ /etc/init.d/globus-ws-java-container restart
$ logout
# su - hoge
$ grid-proxy-init
$ wsrf-query -s ¥
https://tks101.grid.kek.jp:8443/wsrf/services/DefaultIndexService ¥
'/*' | grep 192.168.1.102
```

4.14. WebMDS の設定

・ tks102 上，ユーザ root で行う

```
# ls jakarta-tomcat-5.0.28.tar.gz
  jakarta-tomcat-5.0.28.tar.gz
# tar zxvf jakarta-tomcat-5.0.28.tar.gz
# mv jakarta-tomcat-5.0.28 /usr/local/
# chown -R globus:globus /usr/local/jakarta-tomcat-5.0.28/
# cd /etc/grid-security/
# ln -sf /usr/local/globus-4.2.1/share/certificates certificates
# su - globus
$ vi /usr/local/globus-4.2.1/lib/webmds/conf/indexinfo
```

※下記のように修正

```
  <value>
     https://tks101.grid.kek.jp:8443/wsrf/services/DefaultIndexService
  </value>
$ export CATALINA_HOME=/usr/local/jakarta-tomcat-5.0.28
$ /usr/local/globus-4.2.1/lib/webmds/bin/webmds-create-context-file ¥
$CATALINA_HOME/conf/Catalina/localhost
```

$ $CATALINA_HOME/bin/startup.sh ←一度電源を落とすと手動で動かしてあげないといけない？

```
Using CATALINA_BASE:   /usr/local/jakarta-tomcat-5.0.28
Using CATALINA_HOME:   /usr/local/jakarta-tomcat-5.0.28
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat-5.0.28/temp
Using JAVA_HOME:       /usr/local/java/jdk1.6.0_24
```

※tks001 マシンからブラウザを立ち上げ、http://tks102.grid.kek.jp:8080/webmds/へアクセスし、"A list of resources registered to the local default index service."をクリックして下記のようなリソースが表示されることを確認

ServiceGroup Overview

This page provides a brief overview of Web Services and/or WS-Resources that are members of a WS-ServiceGroup.

This WS-ServiceGroup has 4 direct entries, 7 in whole hierarchy.

Resource Type	ID	Information	
ServiceGroup	192.168.1.102	This WS-ServiceGroup has 0 direct entries, 0 including descendants.	detail
GRAM	192.168.1.101	Multi resource manager	detail
GRAM	192.168.1.101	Fork resource manager with 1 queues, submitting to 0 cluster(s) of 0 host(s).	detail
RFT	192.168.1.101	0 active transfer resources, transferring 0 files. 54.4 KB transferred in 7 files since start of database.	detail
GRAM	192.168.1.102	Fork resource manager with 1 queues, submitting to 0 cluster(s) of 0 host(s).	detail
GRAM	192.168.1.102	Multi resource manager	detail
RFT	192.168.1.102	0 active transfer resources, transferring 0 files. 194.21 KB transferred in 10 files since start of database.	detail

Please report bugs and feature requests into the Globus Bugzilla.

XSLT transformation provided by servicegrouptable.xsl version 1.13.