

# AMGA Metadata catalogue service for managing mass data of high-energy physics

**Geunchul Park,**

Taesang Huh, SoonWook Hwang, Jaehyuck Kwak

KISTI

*2012. 02. 20*



- What is AMGA?
- Why AMGA?
- AMGA Features & AMGA APIs
- Conclusion

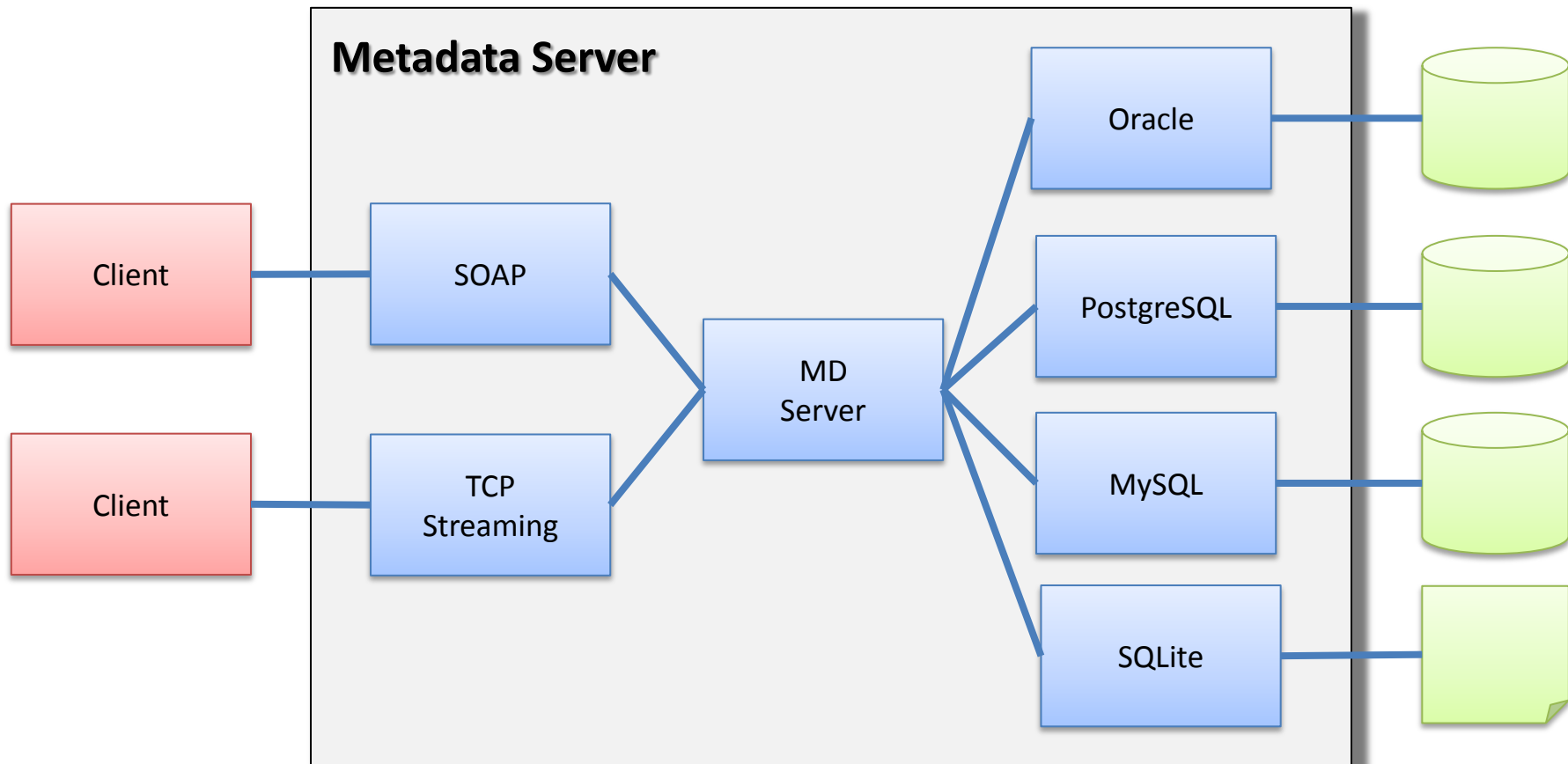
# What is AMGA?

- **AMGA is a metadata service for the Grid**
  - It represents a database access service for Grid applications which allows user, and user jobs to **discovery data describing their files** in order to access them in the appropriate way.
- **Part of gLite middleware and EMI distribution**
- **Requirements from HEP community**
  - Millions of files, 6000+ users, 200+ computing centres
  - Mainly (real-only) file metadata
  - Main concerns : Scalability, Performance, Fault-tolerance,  
Support for Hierarchical Collection
- **Requirements from Biomed community**
  - Smaller scale than HEP
  - Main concerns : Security

## ***Four powerful-features of AMGA***

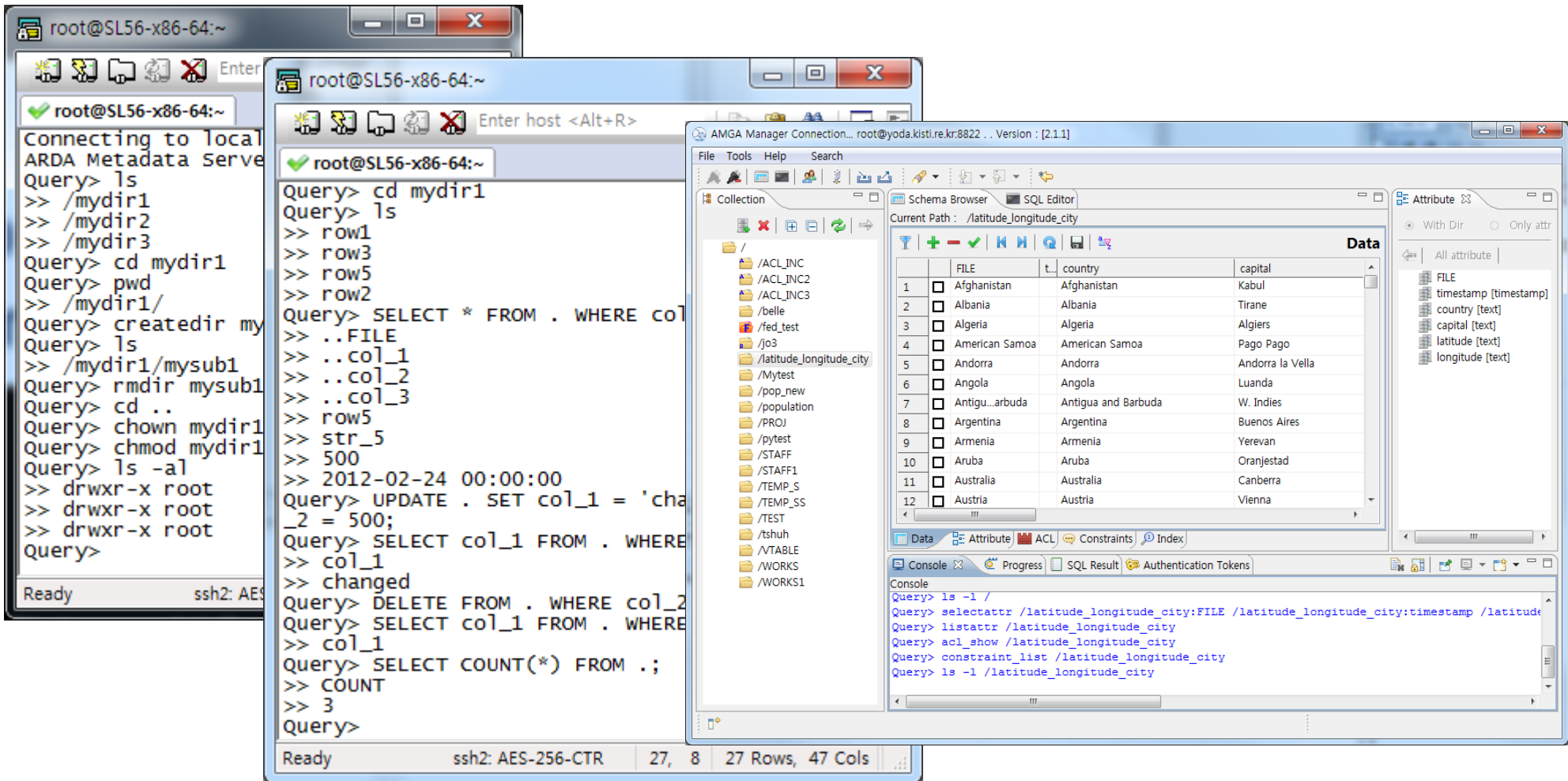
- **AMGA supports the various databases**
  - Oracle, PostgreSQL, MySQL, SQLite
- **AMGA provides easy interfaces**
  - Easy interfaces what look like linux file system, Native SQL and GUI Client
- **AMGA provides two special feature for Grid.**
  - Replication and Federation
- **AMGA provides APIs of the various programming languages**
  - C++, Java, php, Python, shell scripts

- **Two Front-ends : TCP/IP, SOAP**
- **Various Backend : PostgreSQL, Oracle, MySQL, and SQLite**



# AMGA's interfaces

- AMGA Commands, Native SQL, AMGA Manager(GUI)



The image displays three overlapping windows. On the left, a terminal window shows a sequence of shell commands and their outputs, including directory creation, file permissions, and SQL queries. In the center, another terminal window shows a similar sequence of commands, including a file update and a delete operation. On the right, the AMGA Manager GUI is shown, featuring a Schema Browser, a Data table, and a Console window.

**Terminal 1 (Left):**

```
root@SL56-x86-64:~  
Connecting to local ARDA Metadata Server  
Query> ls  
>> /mydir1  
>> /mydir2  
>> /mydir3  
Query> cd mydir1  
Query> pwd  
>> /mydir1/  
Query> createdir mydir1/mysub1  
Query> ls  
>> /mydir1/mysub1  
Query> rmdir mysub1  
Query> cd ..  
Query> chown mydir1  
Query> chmod mydir1  
Query> ls -al  
>> drwxr-x root  
>> drwxr-x root  
>> drwxr-x root  
Query>
```

**Terminal 2 (Center):**

```
root@SL56-x86-64:~  
Query> cd mydir1  
Query> ls  
>> row1  
>> row3  
>> row5  
>> row2  
Query> SELECT * FROM . WHERE col_1 = 'changed'  
>> ..FILE  
>> ..col_1  
>> ..col_2  
>> ..col_3  
>> row5  
>> str_5  
>> 500  
Query> UPDATE . SET col_1 = 'changed', col_2 = 500;  
Query> SELECT col_1 FROM . WHERE col_1 = 'changed'  
>> col_1  
>> changed  
Query> DELETE FROM . WHERE col_2 = 500;  
Query> SELECT col_1 FROM . WHERE col_1 = 'changed'  
>> col_1  
Query> SELECT COUNT(*) FROM .;  
>> COUNT  
>> 3  
Query>
```

**AMGA Manager GUI (Right):**

Schema Browser: /latitude\_longitude\_city

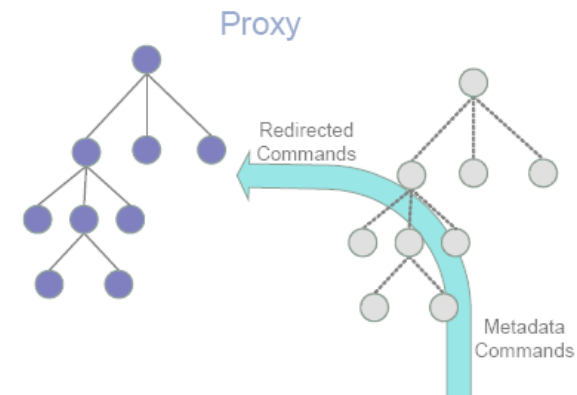
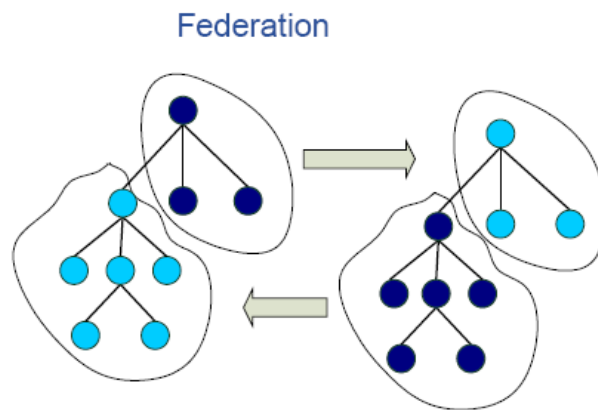
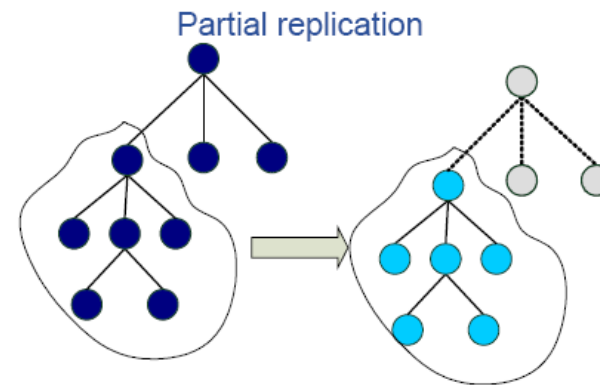
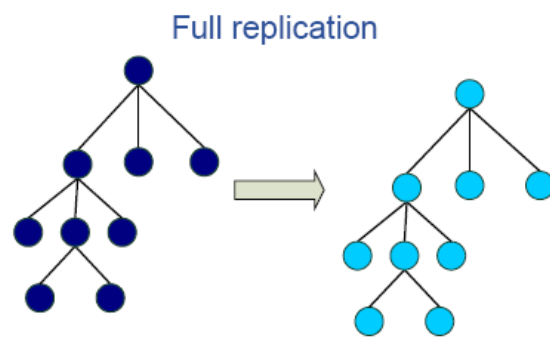
FILE	country	capital
Afghanistan	Afghanistan	Kabul
Albania	Albania	Tirane
Algeria	Algeria	Algiers
American Samoa	American Samoa	Pago Pago
Andorra	Andorra	Andorra la Vella
Angola	Angola	Luanda
Antigua and Barbuda	Antigua and Barbuda	W. Indies
Argentina	Argentina	Buenos Aires
Armenia	Armenia	Yerevan
Aruba	Aruba	Oranjestad
Australia	Australia	Canberra
Austria	Austria	Vienna

Console:

```
Query> ls -l /  
Query> selectattr /latitude_longitude_city:FILE /latitude_longitude_city:timestamp /latitude_longitude_city:country  
Query> listattr /latitude_longitude_city  
Query> acl_show /latitude_longitude_city  
Query> constraint_list /latitude_longitude_city  
Query> ls -l /latitude_longitude_city
```

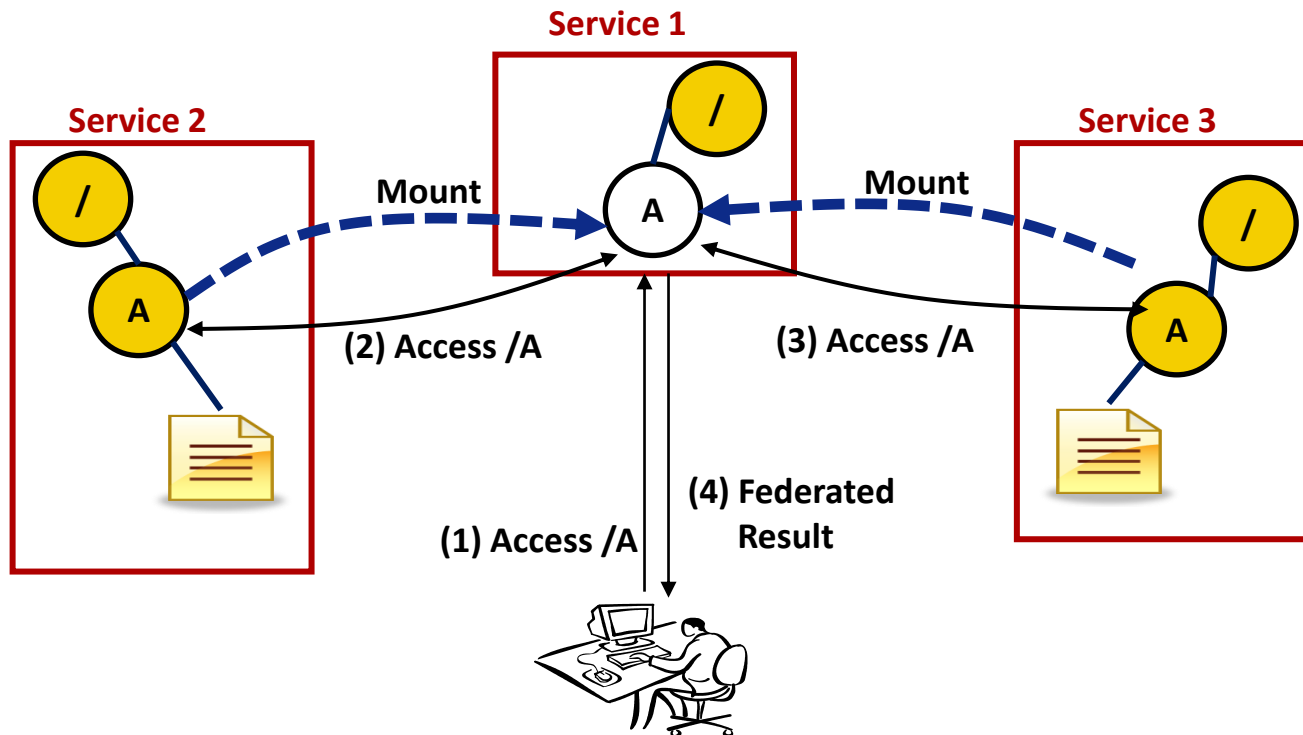
- **Metadata Replication**

- Master-Slave & Asynchronous communication model
- Full & Partial replications allowed



- **Metadata Federation**

- A mechanism to integrate distributed metadata seamlessly.
- provide a user with a virtualized view on metadata as if one metadata server has all data which are actually distributed at multiple sites.





- **AMGA in the Shell Script**

- AMGA provides “mdcli” executable file.
- Using mdcli command in the shell scripts.
- Communicate with TCP/IP directly.

```
#!/bin/sh

MDCLI=mdcli
LSTPX=$(date +%y%m%d_%N)
.....
if [ $ISDIR = 'd' ]
then
  DIR=$(echo ${ENTRY} | sed s/'*'/ ' / | awk '{ s=$2;
print s;}')
  ARG=$2"--"
  echo $2$DIR"/"
  $MDCLI "listattr $DIR" | awk '{ print " a:"$1 }'
  $0 $DIR $ARG
else
.....
```

```
#!/usr/bin/perl -w
# make socket
$sock = IO::Socket::INET->new(
  PeerAddr => $host,
  PeerPort => $port,
  Proto => 'tcp'
);
if (!defined $sock) {
  my $s_str="Failed to connect AMGA Server
($host:$port)";
  comm_nagios(2,$s_str);
}

# Send AMGA Monitoring Command
print $sock "statistics\n\n";
```

- **C++ Client API**
  - Two APIs available(md\_api, MDlient class)
- **C++ Simple Source**

```
#include <MDClient.h>
.....
MDClient client;
client.connectToServer()
.....
comm = "transaction" ;
res=client.execute(comm)
.....
res=client.execute(amgaQuery)
.....
while(!client.eot()) {
    string row;
    if(res=client.fetchRow(row))
.....
```

- **Python API**
  - glite.amga.api-python RPM
- **Python Simple Source**

```
#!/usr/bin/env python
.....
try:
    import mdstandalone
    import mdclient
    import mdparser
    import mdinterface
except ImportError, e:
.....
client = mdclient.MDClient('localhost', 8822, 'guest')
client.createDir("/pytest")
client.cd("/pytest")
client.addAttr(".", "events", "int")
client.addAttr("/pytest", "eventGen", "varchar(20)")
.....
```

- **Java API**
  - RPM and tar ball(glite-amga-api-java.jar)
- **Java Simple Source**

```
import arda.md.javaclient.*;
.....
MDServerConnection serverConn = new MDServerConnection(
MDServerConnectionContext.loadDefaultConfiguration());
MDClient mdClient = new MDClient(serverConn);
.....
AttributeDef[] attrs = mdClient.listAttr("/test");
.....
String[] keys = {"gen", "events"};
NamedAttributesIterator attrs = mdClient.getAttr("/test", keys);
.....
while (attrs.hasNext()) {
    NamedAttributes entry = attrs.next();
    System.out.println("File: " + entry.getEntryName());
}
.....
```

- **PHP API**

- mdclient.php

- **PHP Simple Source**

```
include 'mdclient.php';  
.....  
$client = new gLibrary($host, $port, $login);  
$client->requireSSL("x509_xxx", "x509_xxx");  
$client->connect();  
.....  
$attrs = array("/gLibTest/Entries:FileName", "Size");  
$condition = "Size > 1000000 order(FileName)";  
$num_attrs = count($attrs);  
$result = $client->selectAttr($attrs, $condition);  
while (!$client->eot()) {  
    $FileName = $client->getSelectAttrEntry();  
    $Size = $client->getSelectAttrEntry();  
    echo $FileName[0]. " (". $Size[0]. ")<br>";  
}  
.....
```

- **AMGA is the best solution of a metadata catalogue service for HEP.**
  - AMGA supports the various databases
  - AMGA provides easy interfaces
  - AMGA provides two special feature for Grid.
  - AMGA provides APIs of the various programming languages
- **You can use AMGA.**
  - AMGA's Official site : <http://amga.web.cern.ch/amga/>
  - Contact to KISTI
    - SoonWook Hwang([hwang@kisti.re.kr](mailto:hwang@kisti.re.kr))
    - Taesang Huh([tshuh@kisti.re.kr](mailto:tshuh@kisti.re.kr))
    - Geunchul Park([gcpark@kisti.re.kr](mailto:gcpark@kisti.re.kr))
    - Jaehyuck Kwak([jhkwak@kisti.re.kr](mailto:jhkwak@kisti.re.kr))



Thank you