NDGF
Nordic DataGrid Facility

Information systems in the NORDICS

Vera Hansper/CSC, NDGF
Ulf Tigerstedt/CSC, FGI
Kalle Happonen/CSC, FGI

CSC

NDGF
Nordic DataGrid Facility

ARC Information system

- **ARC Information service is BDII based**
  - Scalable, dynamic, robust
  - Gory details in http://www.nordugrid.org/documents/arc_infosys.pdf
  - Lists of "NorduGrid GIISes" are generally distributed with the middleware – they are compiled into the client.
  - Sites locally configure their files and register information to the index[1-4].nordugrid.org servers
    - /etc/arc.conf

CSC

http://www.nordugrid.org/documents/arc_infosys.pdf

Figure 1 presents an overview of the ARC information system components.

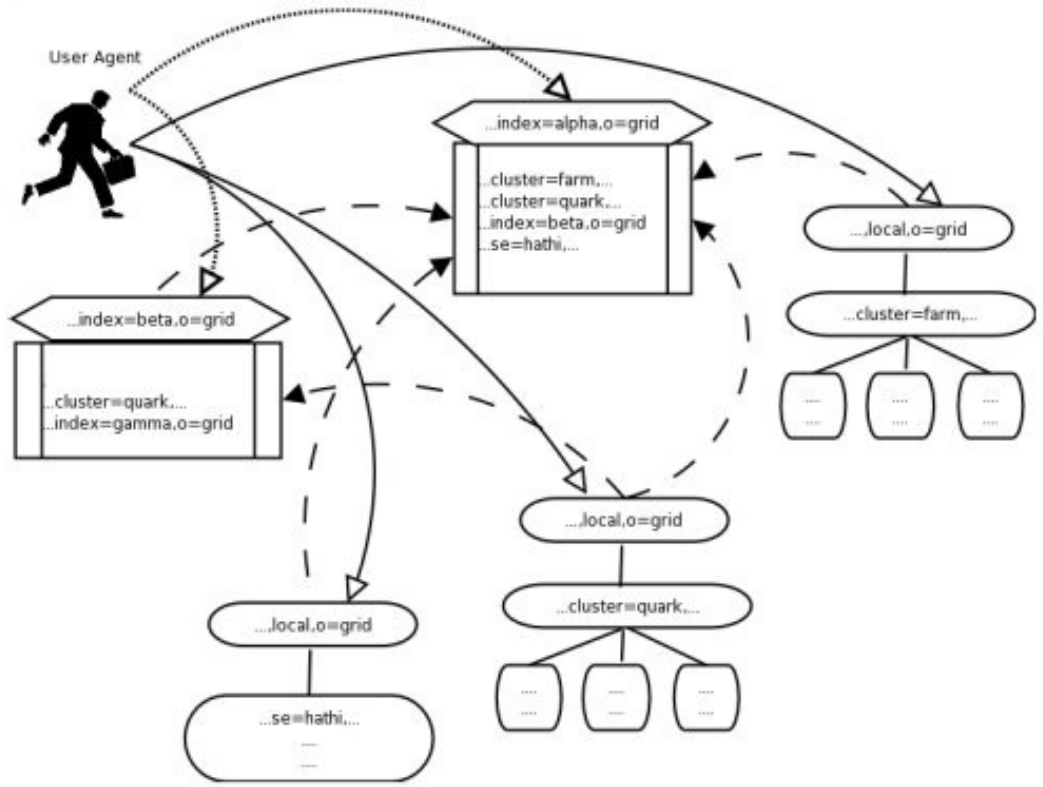


Figure 1: Overview of ARC information system components.

NDGF
Nordic DataGrid Facility

Issues

- **ARC BDII and gLite BDII don't talk the same language**
  - **(NOTE: Some sites use an OLD version of ARC (0.8))**
  - Both use ldap [check]
    - New versions of ARC use XML
  - Both use Glue Schema [NO]
    - ARC uses nordugrid schema
    - gLite uses 1.3
- THESE ARE INCOMPATIBLE!

CSC

NDGF
Nordic DataGrid Facility

List of Nordic CEs and ARC versions

arc-ce01.pdc.kth.se	: nordugrid-arc-0.8.1.1
korundi.grid.helsinki.fi	: nordugrid-arc-0.8.3
jade-cms.hip.fi	: nordugrid-arc-0.8.3
mon01.grid.lumii.lv	: nordugrid-arc-0.8.3.1
norgrid.bccs.uib.no	: nordugrid-arc-0.8.3.1
arc.bccs.uib.no	: nordugrid-arc-0.8.3.1
ce02.titan.uio.no	: nordugrid-arc-1.0.1
usva.fgi.csc.fi	: nordugrid-arc-1.1.0
vuori-arc.csc.fi	: nordugrid-arc-1.1.0
gtpps.csc.fi	: nordugrid-arc-1.1.0
arc-ce.smokerings.nsc.liu.se	:nordugrid-arc-1.1.0
ce01.titan.uio.no	: nordugrid-arc-1.1.0
gateway01.dcsc.ku.dk	: nordugrid-arc-1.1.0
jeannedarc.hpc2n.umu.se	: nordugrid-arc-1.1.0
grad.uppmax.uu.se	: nordugrid-arc-1.1.0
siri.lunarc.lu.se	: nordugrid-arc-1.1.0
gateway03.dcsc.ku.dk	: nordugrid-arc-1.1.0

CSC

NDGF
Nordic DataGrid Facility

Solution at Nordic Level

- **Map ARC information and convert it to glue 1.3**
  - gLite Site BDII can then pull the data from the ARC resources
  - All ARC CEs now have this conversion “file”
- **Was supposed to be a short term solution**
  - With a long life ...
- **BUT IT'S ALSO MESSY**
  - And not very scalable in the long run

NDGF
Nordic DataGrid Facility

glite-info-provider-ndgf.txt

```
#!/usr/bin/perl -w
# Queries NDGF site information and translates it from ARC to GLUE schema
# Prototype by L.Field (2006)
# Corrections and Performance enhancements by M.Flechl (2006-08-17)
# Merged with adjusted bdii-update (David Groep, L.Field, M.Litmaath) by L.Field and M.Flechl (2006-08-21)
# ARC GIIS traversal and some sanity checks for failing GRISes by Mattias Wadenstein (2007-07-03)
# $Id: glite-info-provider-ndgf,v 1.1 2006/08/30 12:13:15 lfield Exp $

use strict;
use LWP::Simple;
use POSIX;
use IO::Handle;

sub translator;

my $tmp_dir = "/tmp/gip";
# "nordugrid" for the general arc bdii, "ndgf" for bdii.ndgf.org
my $usemode = "ndgf";
my $recursion = 1;
my $index_urls = undef;
my @grises;

$usemode = $0;

$usemode =~ s/.*glite-info-provider-//;
$tmp_dir = $tmp_dir . '/' . $usemode;
`mkdir -p $tmp_dir  unless -d $tmp_dir;
my $log_file = $tmp_dir."/glite-info-provider-ndgf.log";

if($usemode eq "ndgf") {
        $index_urls = "ldap://index1.ndgf.org:2135/mds-vo-name=ndgf,o=grid ldap://index2.ndgf.org:2135/mds-vo-name=ndgf,o=grid";
} elsif ($usemode eq "se-snic-t2") {
        $index_urls = "ldap://index1.ndgf.org:2135/mds-vo-name=se-snic-t2,o=grid ldap://index2.ndgf.org:2135/mds-vo-name=se-snic-t2,o=grid";
} elsif ($usemode eq "lmcsul") {
        $grises{"mon01.grid.lumii.lv-LMCSUL ldap://mon01.grid.lumii.lv:2135/nordugrid-cluster-name=mon01.grid.lumii.lv,Mds-Vo-name=local,o=grid"}=1;
} elsif ($usemode eq "no-norgrid-t2") {
        $index_urls = "ldap://index1.ndgf.org:2135/mds-vo-name=no-norgrid-t2,o=grid ldap://index2.ndgf.org:2135/mds-vo-name=no-norgrid-t2,o=grid";
} else {
        $index_urls = "ldap://index1.nordugrid.org:2135/mds-vo-name=nordugrid,o=grid ldap://index2.nordugrid.org:2135/mds-vo-name=nordugrid,o=grid ldap://ind
ex3.nordugrid.org:2135/mds-vo-name=nordugrid,o=grid ldap://index4.nordugrid.org:2135/mds-vo-name=nordugrid,o=grid";
}
my $search_timeout = "30";
my $newConfig;
my $msg = "GOT TIRED OF WAITING";
my $old_sel;
my $sitename=uc $usemode;
$sitename='NDGF-T1' if $sitename eq 'NDGF';
my %pledge;
$pledge{"NDGF-T1,alice"} = 12535;
$pledge{"NDGF-T1,atlas"} = 13229;
$pledge{"SE-SNIC-T2,alice"} = 2820;
$pledge{"SE-SNIC-T2,atlas"} = 5050;
$pledge{"NO-NORGRID-T2,alice"} = 0;
$pledge{"NO-NORGRID-T2,atlas"} = 2305;

open(OLDOUT, ">&STDOUT");
open(STDOUT,"> $log_file") || die "Can't redirect STDOUT to $log_file!\n";
STDOUT->autoflush(1);

my @urls;     # An arry of URLS that return ldif.
my @giis;

#Ensure that the temp directory exists and is empty
`mkdir -p $tmp_dir/giis`;
`mkdir -p $tmp_dir/giis_old`;
`rm -f $tmp_dir/*.ldif`;
`rm -f $tmp_dir/*.org`;
`rm -f $tmp_dir/cache`;
#`rm -f $tmp_dir/*.log`;
```

```
my @pid;      # The pids of the forked processes.
my $pid;      # A pid for one process.
my $region;   # A region name for the ldif source.
my $command;  # A command that will return ldif.

if(defined $index_urls) {
print STDOUT "Searching $index_urls\n";
print STDOUT "Waiting $search_timeout s for query results.\n\n";

@giis = split / /,$index_urls;

#Loop through for each giis source for grises, until done
while(@giis) {
foreach(@giis){

    if(m/ldap:/){
        #Split the information from the url.
        if (m|^(ldap):.//(.+):([0-9]+)/(.+)|) {
            $region=$1;
            $command="ldapsearch -x -LLL -h $2 -p $3 -b \'$4\' -s base giisregistrationstatus"
                . " | perl -p -00 -e \"s/\r\n //g; s/\n //g\""
                . "> $tmp_dir/giis/$2.ldif 2>&1";
            print STDOUT "Querying $2:\n";
                print STDOUT "$command\n";

        } else {
            print STDOUT "ignoring badly formatted line: '$_'\n";
            next;
        }
    }

    # Fork the search.
     if (!defined($pid=fork)) {
        print STDOUT "cannot fork: $!\n";
        next;
    }

    unless ($pid) {
        # Set our process group to a distinct value.
        setpgrp();
        my $msg = "GOT TIRED OF WAITING";

        # Eval will kill the process if it times out.
        eval {
            local $SIG{ALRM} = sub { die "$msg" };
            alarm ($search_timeout);  #Will call alarm after the timeout.

            if(system("$command")){
                print STDOUT "$2: ";
                system("cat $tmp_dir/giis/$2.ldif")
                }
            alarm(0); # Cancel the pending alarm if responds.
        };

        # This sections is executed if the process times out.
        if ($@ =~ /$msg/) {
            `rm -f $tmp_dir/giis/$2.ldif`;
            print STDOUT "\n$2: Timed out";
            my $PGRP=getpgrp();
            kill (-SIGKILL(), $PGRP);
            print STDOUT "\n";
            exit 1;
        }
        exit 0;
    }
    push @pid, $pid;
}
```

NDGF
NORDIC DATAGRID FACILITY

glite-info-provider-ndgf.txt

```
foreach(@pid){
    waitpid($_, 0);
}
my %new_giises=();
open(P,"cat $tmp_dir/giis/*.ldif |");

$old_eol=$/;
$/ = "\n\n";
# loop over all records
while (<P>) {
        my %record = ();
        my @lines = split /[\n]+/;
        my $url;
        foreach (@lines) {
                my $key;
                my $value;
                ($key,$value)=split /: /;
                $record{$key}=$value;
        }
        if (defined $record{'Mds-Service-Ldap-suffix'} and not ($record{'Mds-Service-Ldap-suffix'}  =~ /Mds-Vo-name=local/)) {
                # is a giis
                next unless defined $record{'Mds-Service-hn'};
                next unless defined $record{'Mds-Service-port'};
                next unless defined $record{'Mds-Service-Ldap-suffix'};
                next if defined $record{'Mds-Reg-status'} && $record{'Mds-Reg-status'} eq 'PURGED';
                $url = "ldap://$record{'Mds-Service-hn'};$record{'Mds-Service-port'}/$record{'Mds-Service-Ldap-suffix'}";
                print STDOUT "new giis: $url\n";
                $new_giises{$url}=1 if $recursion;
        } elsif (defined $record{'dn'} and $record{'dn'} =~ /nordugrid-cluster-name=([^,]+), *Mds-Vo-name=([^,]+), *o=grid/i) {
                my $name="$1-$2";
                # is a CE gris
                next unless defined $record{'Mds-Service-hn'};
                next unless defined $record{'Mds-Service-port'};
                next unless defined $record{'Mds-Service-Ldap-suffix'};
                next if defined $record{'Mds-Reg-status'} && $record{'Mds-Reg-status'} eq 'PURGED';
                $url = "$name ldap://$record{'Mds-Service-hn'};$record{'Mds-Service-port'}/$record{'Mds-Service-Ldap-suffix'}";
                print STDOUT "new gris: $url\n";
                $grises{$url}=1;
        }
   }
    mv $tmp_dir/giis/*.ldif $tmp_dir/giis.old/`;
   @giis =(keys %new_giises);
}
$/=$old_eol;
}
@urls = keys %grises;
print STDOUT "Waiting $search_timeout s for query results.\n\n";

#Loop through for each ldif source
foreach(@urls){

    if(m/ldap:/){
        #Split the information from the url.
        if (m|^([-.\w]+)\s+ldap://(.+):([0-9]+)/(.+)|) {
            $region=$1;
            $command="ldapsearch -x -LLL -h $2 -p $3 -b $4 '(|(objectClass=nordugrid-queue)(objectClass=nordugrid-cluster))'"
                . "> $tmp_dir/$region.ldif 2>&1";
            print STDOUT "Querying $region\n";

        } else {
            print STDOUT "ignoring badly formatted line: '$_'\n";
            next;
        }
    }

    # Fork the search.
     if (!defined($pid=fork)) {
        print STDOUT "cannot fork: $!\n";
        next;
    }

    unless ($pid) {
        # Set our process group to a distinct value.
        setpgrp();
        my $msg = "GOT TIRED OF WAITING";

        # Eval will kill the process if it times out.
        eval {
            local $SIG{ALRM} = sub { die "$msg" };
            alarm ($search_timeout);  #Will call alarm after the timeout.

            if(system("$command")){
                print STDOUT "$region: ";
                system("cat $tmp_dir/$region.ldif")
                }
            alarm(0); # Cancel the pending alarm if responds.
        };

        # This sections is executed if the process times out.
        if ($@ =~ /$msg/) {
            `rm -f $tmp_dir/$region.ldif`;
            print STDOUT "\n$region: Timed out";
            my $PGRP=getpgrp();
            kill (-SIGKILL(), $PGRP);
            print STDOUT "\n";
            exit 1;
        }
        exit 0;
    }
    push @pid, $pid;
}

foreach(@pid){
    waitpid($_, 0);
}

#STDOUT->autoflush(1);
print STDOUT "\n\nDoing Translation from ARC to Glue\n";

####Global variables for translator
##################################
use vars qw($DEFAULT); $DEFAULT = -1;
use vars qw($outbIP $inbIP $glueSubClusterUniqueID $norduBenchmark $norduOpsys $norduNodecput $norduNodecpu);
use vars qw($glueHostMainMemoryRAMSize $glueHostArchitecturePlatformType $glueSubClusterUniqueID $GlueHostBenchmarkSI00 $GlueHostBenchmarkSF00);
use vars qw($glueSubClusterName $glueSubClusterPhysicalCPUs $glueClusterUniqueID $glueCEUniqueID $GlueHostProcessorOtherDescription);
use vars qw($AssignedSlots $mappedStatus $waitingJobs $totalJobs $waitingJobs $freeSlots $estRespTime $AtlasShare $AliceShare);
use vars qw(%attr @envs);
use vars qw($totalcpus $hepspec $hepspecsum @lines);
##################################


$hepspecsum=0;
$hepspec=5;
$totalcpus=1;
my @files=glob("$tmp_dir/*.ldif");
foreach(@files){
        open FH,$_ or die "$_: $!";
        while(<FH>) {
                $totalcpus = $1 if $_ =~ /nordugrid-cluster-totalcpus: (\d+)/;
                $hepspec = $1 if $_ =~ /nordugrid-cluster-benchmark: HEPSPEC2006 @ (\d+[\d,]+)/;
        }
        $hepspecsum+=$totalcpus*$hepspec;
}

$AtlasShare = int(100*$pledge{$sitename . ".atlas"} / $hepspecsum);
$AliceShare = int(100*$pledge{$sitename . ".alice"} / $hepspecsum);

$totalcpus=$pledge{$sitename . ".atlas"};
```

NDGF
Nordic DataGrid Facility

And then other hacks

- **The Site BDII was hacked to ensure that information from the ARC sites is transferred upstream**
  - These have been applied to gLite Site BDII in general

```
*** /usr/sbin/bdii-update       2011-04-15 11:57:09.000000000 +0300
--- /usr/sbin/bdii-update.bak   2011-03-29 19:51:10.000000000 +0300
***************
*** 521,534 ****
                if 'mds' in map( lambda x : x.lower(),
entry['objectclass']):
                    if 'gluetop' in map( lambda x : x.lower(),
entry['objectclass']):
                        value=dn[12:dn.index(",")]
                        entry = { 'dn': [dn], 'objectclass': ['MDS'],
'mds-vo-name': [value] }
+               if 'MdsVo' in  entry['objectclass']:
+                   entry['objectclass'].remove('MdsVo')
+               if 'mds-validfrom' in entry:
+                   del entry['mds-validfrom']
+               if 'mds-validto' in entry:
+                   del entry['mds-validto']
            entry = convert_back(entry)
            append(entry)
        response = "".join(response)
        return response
--- 521,528 ----
```

- **But, it's still a hack**

CSC

NDGF
Nordic DataGrid Facility

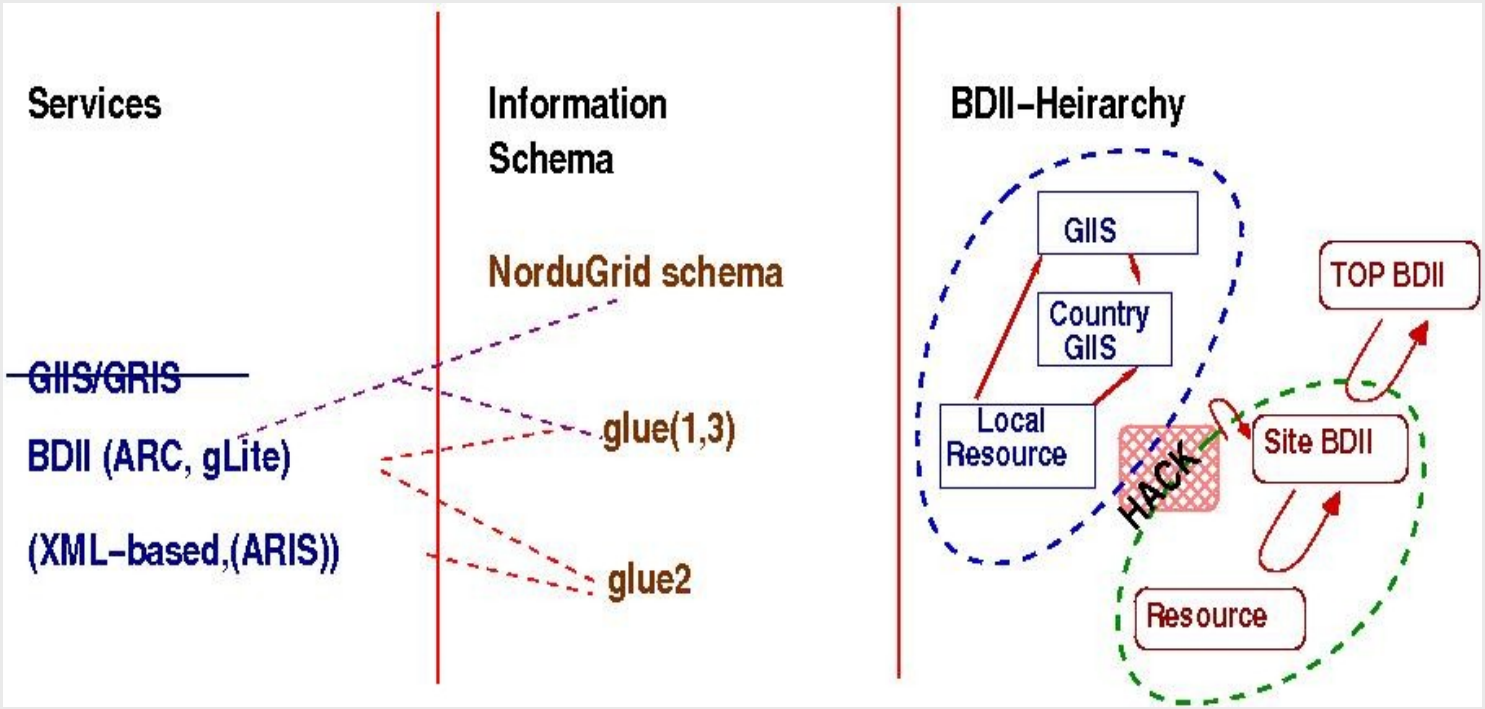
Why not just use gLite BDII?

- **Complex to install**
  - It's got to be done “just right” or it doesn't work
  - Comes with a lot of excess “baggage” that is neither required for functionality nor wanted
    - [above was at least true about one year ago]
- **Why not have gLite use the NorduGrid schema**
  - Again, unreasonable, and would necessitate a lot of work

CSC

Nordic TOP BDII?

- **There is no concept of a TOP BDII in ARC**
  - (see also the previous diagram!)



NDGF
Nordic DataGrid Facility

TOP BDII and Nordics

- **How does the EGI TOP BDII service see Nordic services?**
  - Information Services from arc sites is recorded in the GOCDB **via** the local site BDII

  i.e. for the Finnish NGI at the CSC site – the entry in the GOCDB

  "GIIS URL"[1] **ldap://site-bdii.fgi.csc.fi:2170/mds-vo-name=CSC,o=grid**

- **TOP BDII pulls understands the "GIIS URL", and so information is ported to the TOP BDII**
  - **It doesn't matter which one**
  - **ARC doesn't have worker nodes and WMS like gLite, so this information is primarily used to populate other services like GSTAT and to assist in accounting.**

1: "GIIS" is a misleading term in the GOCDB!

CSC

NDGF
Nordic DataGrid Facility

Solutions?

- **Use a generally acceptable schema!**
  - **GLUE 2**
- **Make the Site and TOP BDII middleware agnostic**
  - **Then these can be used generically across all platforms**
  - **Note that ARC doesn't have the concept of a site or TOP BDII …**
- **OTHER services which rely and use the information from the BDIIs should also be suitably modified**
  - Gstat, accounting, …
  - Will ensure that there is less confusion
- **Migration will be slow :(**

CSC

...

- **GLUE 2 schema is already being incorporated into ARC servers**
  - **Due for release soon?**
- **ARC Clients**
  - **When?**
- **gLite systems?**
  - **When?**