How to download and use ILDG configurations

T. Yoshie

JLDG team

Jan. 27 2010 @ JLDG10

General Procedure

- ✓ **join the ILDG Virtual Organization (VO)**
- ✓ **find ensembles**
- **check access policy**
  - data are public / negotiable / restricted
  - contacting the collaboration is the best way

- **download configurations**
  - **ILDG client tools**
  - RG supports different methods (uberftp, ltools, DiGS tools ...)
- **use configurations for measurements**
  - **file and binary data format**

- **do research and write a paper**
- **acknowledge the collaboration and the ILDG**
  - cite papers specified by the collaboration
  - cite http://www.lqcd.org/ildg

ILDG client tools

- tool kit for download and use ILDG configurations
- developed by Middleware working group
- packed for LatFor datagrid
  - includes packages both for download/upload
  - but, can install only packages necessary for download and use
  - install document given by USQCD grid
  - http://www.usqcd.org/ildg/ildg-client-install.html
- easy to install
  - RPM packages for various Linux OS
  - needs no root privilege with lrpm (localized RPM)
  - needs about 30 min
- what you get
  - globus tools (grid-proxy-init ….)
  - ildg-get to download configurations or metadata
  - utilities (ildg_cksum, LIME)

download configuration

- **You get (a list of) LFN’s from portals or tools**
- **generate proxy certificate (valid for half a day (default))**

```
% grid-proxy-init
Your identity: /C=JP/O=KEK/OU=CRC/OU=Tsukuba/CN=Tomoteru Yoshie
Enter GRID pass phrase for this identity:
Creating proxy ..................................... Done
```

- **invoke ildg-get**

  (download will complete without any message)

```
% ildg-get  lfn://JLDG/CP-PACS/RCNF2/RC12x24-B1800…..
```

- **Note on grid certificate**
  - ILDG VO members can access all RG’s (use official grid certificate)
  - access is restricted to JLDG when you use JLDG private certificate

    put JLDG CA certificate 9fac2951.*

    in $LROOT/etc/grid-security/certificates

more on ildg-get

- You can get ensemble/config XML documents

```
% ildg-get --mdc-only mc://JLDG/CP-PACS+JLQCD/RCNF2+1/RC16x32_B1800....
% ildg-get –mdc-only  lfn: //JLDG/CP-PACS+JLQCD/RCNF2+1/RC16x32_B1800....
```

  – type “ildg-get --help” for options

- debug option is useful (and interesting)

```
% ildg-get --debug=3 lfn://JLDG/CP-PACS/RCNF2/RC12x24-
                                        B1800K014640C1600-D-01420
[main|05:13:04|INFO ] Connecting to FC of regional grid "JLDG“
[main|05:13:05|DEBUG] SURL = gsiftp://www.jldg.org/gfarm/public/ILDG/JLDG/
  CP-PACS/RCNF2/RC12x24-B1800/K014640/RC12x24-B1800K014640C1600-
  D-01420
[main|05:13:05|DEBUG] Execute command: globus-url-copy –dbg
 gsiftp://www.jldg.org/gfarm/public/ILDG/JLDG/CP-PACS/RCNF2/RC12x24-
 B1800/K014640/......
```

file format

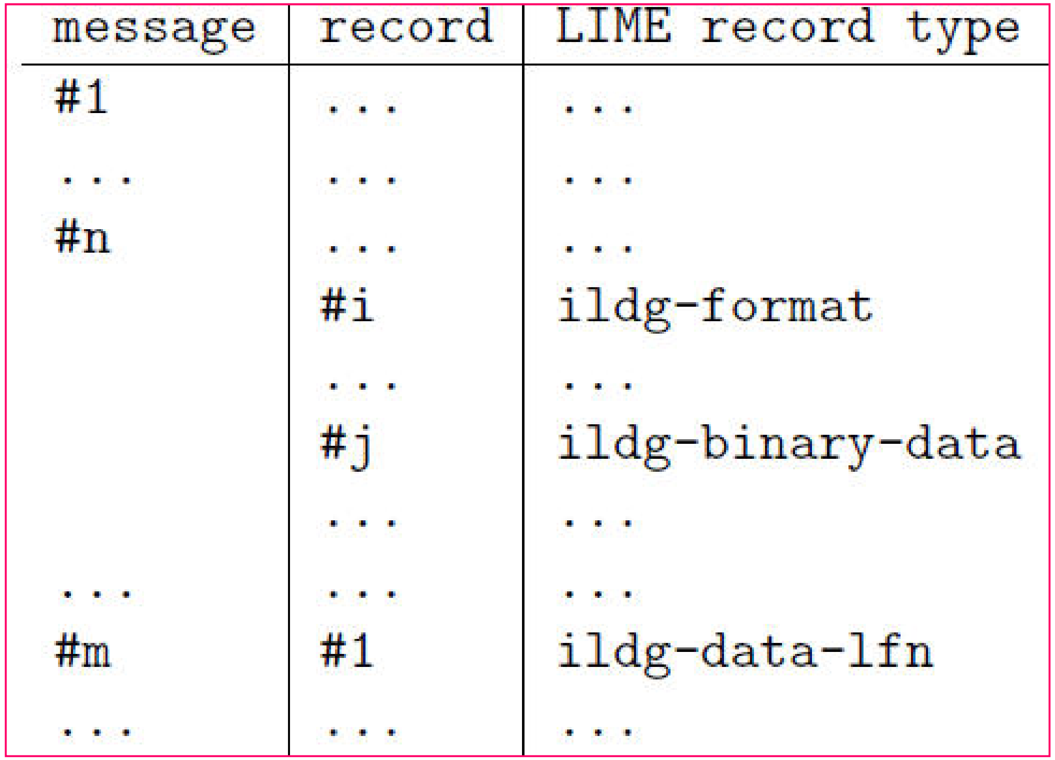
- **ILDG config binary data and other materials are packed with LIME to one file**

- **LIME**
  - Lattice QCD Interchange Message Encapsulation (B.Joo, C.DeTar)
  - a lime file consists of one or more messages
  - a message consists of one or more records
  - a record can be either ASCII or binary data
  - a record has a name “LIME record type”
  - includes header and padding

- **ILDG configuration file should satisfy ILDG file format specification**

http://www-zeuthen.desy.de/~pleiter/ildg/#filefmt

file format: ILDG specification



message	record	LIME record type
#1	...	...
...	...	...
#n	...	...
	#i	ildg-format
	...	...
	#j	ildg-binary-data
	...	...
...	...	...
#m	#1	ildg-data-lfn
...	...	...

- group can insert their own messages/records
  - (type ildg-* is reserved)
- order of #n and #m is not fixed, while #i < #j

file format: ildg-format, ildg-data-lfn

- **ildg-format: minimal info. to read config**

```
<?xml version="1.0" encoding="UTF-8"?>
<ildgFormat xmlns="http://www.lqcd.org/ildg"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://www.lqcd.org/ildg
            http://www.lqcd.org/ildg/filefmt.xsd">
  <version> 1.0 </version>
  <field> su3gauge </field>
  <precision> 64 </precision>
  <lx> 32 </lx> <ly> 32 </ly> <lz> 32 </lz> <lt> 64 </lt>
</ildgFormat>
```

- **ildg-data-lfn: logical file name**
  - enables us to back track

file format: extract binary and check

- **LIME commands**
  - lime_contents <lime_file>
  - lime_extract_type <lime_file> <lime_type>

```
% lime_extract_type ILDG-LIME-config  ildg-binary-data > binary
```

- **checking data integrity**
  - 32 bit CRC

```
% cksum binary
    2685385332 1207959552
        should agree with <crcCheckSum> in the config XML
% ildg_cksum ILDG-LIME-config
    2685385332
```

  - plaquette value (sample program will be provided)

    should agree with <avePlaquette> in the config XML

file format: LIME API

- **you can write a code to read config directory from the LIME packed config file, using LIME API**

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include "lime.h"

#define NS  12
#define NT  24
#define SIZE (NS*NS*NS*NT*4*3*3*16)

unsigned char conf[SIZE] ;
```

```
main(int argc,char **argv) {

  FILE *fp ;
  LimeReader *reader ;
  n_uint64_t nbytes ;

  fp = fopen(argv[1], "r") ;
  reader = limeCreateReader(fp) ;

  do { limeReaderNextRecord(reader) ; }
  while (strncmp(limeReaderType(reader),
           "ildg-binary-data", 16)) ;

  nbytes = limeReaderBytes(reader) ;
  limeReaderReadData(conf, &nbytes, reader) ;

  limeDestroyReader(reader) ;
  fclose(fp) ;
}
```

Binary format

- **ildg-binary-data**
  - sequence of floats/doubles

$$\chi^{+}(n)U_{\mu}(n)\chi(n+\hat{\mu}) = \sum_{a,b=0}^{2}[\chi^{+}(n)]_{a}[U_{\mu}(n)]_{ab}[\chi(n+\hat{\mu})]_{b}$$

In C format: U[t][z][y][x][mu][a][b][ri]

ri=0,1 for real/imaginary parts

a,b=0,1,2

mu=0 (x), 1 (y), 2 (z), 3 (t)

x=0,1.... NX-1 (NX: size in x direction)

In Fortran format: U(ri,b,a,mu,x,y,z,t)

color indices are transposed in Fortran

  - floats/doubles are coded in big endian byte order
  - Fortran program needs a device, because of no header

    sample program will be provided