

Understanding LSST I/O patterns

status and perspectives

Fabio Hernandez
fabio@in2p3.fr

LSST France calcul, Lyon, May 29th, 2015

Goal

- To understand **file I/O patterns** induced by LSST stack
- Ultimate goal: to identify the requirements a storage platform needs to satisfy to support LSST workflows
alternatively, to adapt LSST software to take into account the intrinsic limitations of available storage platforms

Status

- Developed clueFS, a FUSE-based synthesised file system to **intercept**, **trace** and **forward** I/O operations to the underlying file system

this is the tool we use to collect the traces

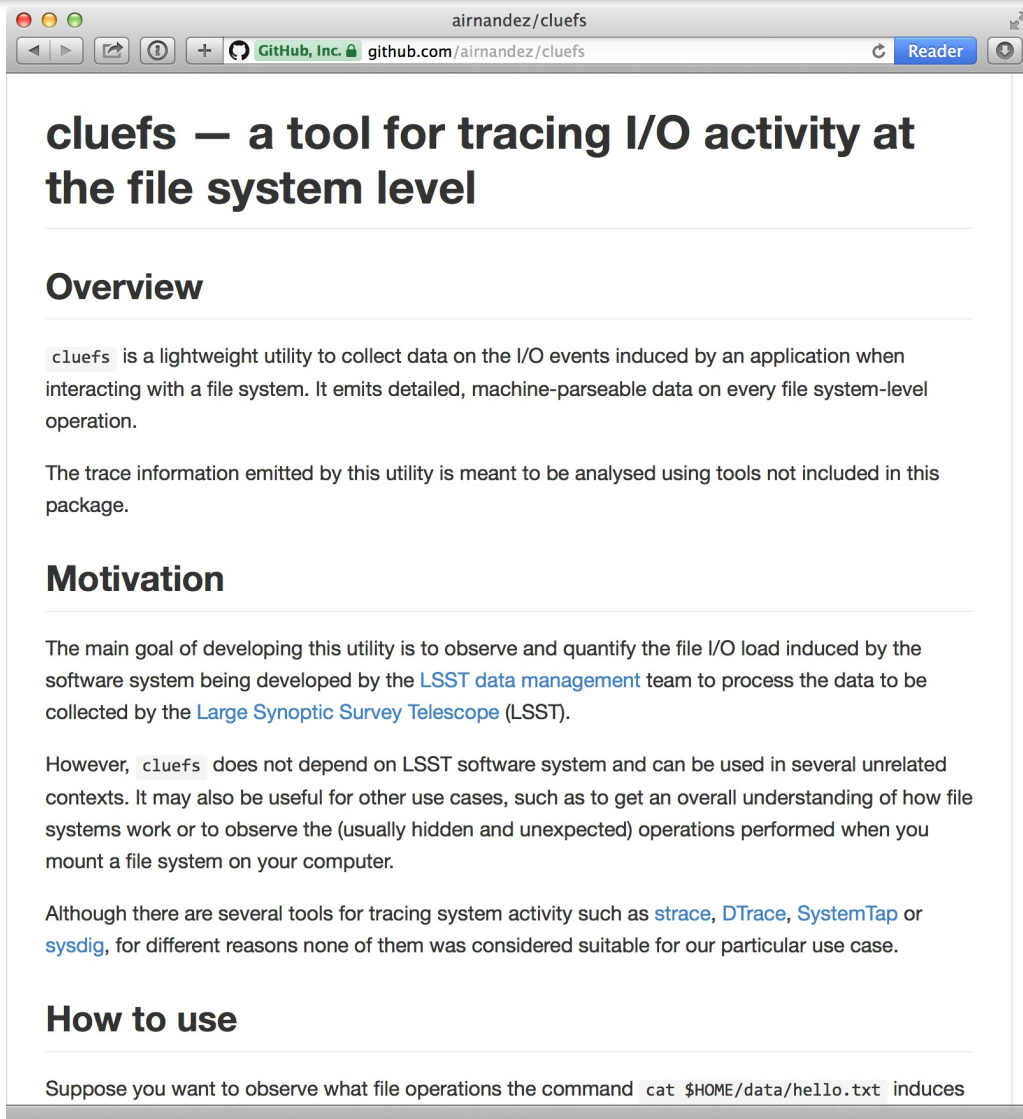
- Inspired by tools such as strace, SystemTap, sysdig, dtrace, ...

- Simple to use

no super-user privileges

Linux and MacOS

easy-to-install stand-alone executable



<https://github.com/airnandez/cluefs>

Example

Mount the file system

```
$ cluefs --mount=/tmp/trace --shadow=$HOME/data &
```

Observe what **cat** command induces:

```
$ cat /tmp/trace/hello.txt
```

...

```
2015-03-23T10:26:35.839367864Z,2015-03-23T10:26:35.839794442Z,426578,fabio,9986,lsst,1021,/usr/bin/cat,23161,/home/fabio/data/hello.txt,stat
```

```
2015-03-23T10:26:35.840322045Z,2015-03-23T10:26:35.840364156Z,42111,fabio,9986,lsst,1021,/usr/bin/cat,23161,/home/fabio/data/hello.txt,openfile,O_RDONLY,0000
```

```
2015-03-23T10:26:35.840556082Z,2015-03-23T10:26:35.840572507Z,16425,fabio,9986,lsst,1021,/usr/bin/cat,23161,/home/fabio/data/hello.txt,read,15,0,4096,15
```

```
2015-03-23T10:26:35.841009818Z,2015-03-23T10:26:35.901634332Z,60624514,fabio,9986,lsst,1021,/usr/bin/cat,23161,/home/fabio/data/hello.txt,flush
```

```
2015-03-23T10:26:35.90204842Z,2015-03-23T10:26:35.902054482Z,6062,root,0,root,0,,0,/home/fabio/data/hello.txt,close
```

...

Analysis

- Currently developing the Python notebooks to summarise the traces collected by executing the LSST demo

example of preliminary findings: a single index file of astrometry.net is open 210 times when executing the stack demo

not necessarily an anomaly, but an interesting fact we could exploit

- Once this is done, we will do the same exercise with more realistic workflows developed by Dominique Boutigny, using the stack and CFHT data

Analysis (cont.)

LSST-IO-Analysis

IPy localhost:8888/notebooks/LSST-IO-Analysis.ipynb

LSST-France calcul — Atelier logiciel LSST (29 May 2015)

Home

LSST-IO-Analysis

IP[y]: Notebook LSST-IO-Analysis Last Checkpoint: May 29 09:58 (autosaved)

File Edit View Insert Cell Kernel Help

Markdown Cell Toolbar: None

Import data into Pandas

In this step we use Pandas to read the CSV-formatted raw data and create a data frame.

```
In [7]: # Import the data file and show its first entries
column_names=[
    "Start", "End", "Duration",
    "UserName", "UserId", "GroupName", "GroupId", "ProcessName", "ProcessId",
    "Path", "ObjType",
    "Operation", "OpArg1", "OpArg2", "OpArg3", "OpArg4", "OpArg5"
]
frame = pd.read_csv("./ubuntu-14.04-lsst_dm_stack_demo.csv", header=None, names=column_names, parse_dates=False)
frame.head(10)
```

```
Out[7]:
```

	End	Duration	UserName	UserId	GroupName	GroupId	ProcessName	ProcessId	Path
36054101Z	2015-04-28T13:49:27.336139145Z	85044	ubuntu	1000	ubuntu	1000	/bin/bash	2224	/home/ubuntu/lsst/lsst_dm_stack_demo
37513981Z	2015-04-28T13:49:28.887592037Z	78056	ubuntu	1000	ubuntu	1000	/bin/bash	2224	/home/ubuntu/lsst/lsst_dm_stack_demo
37903015Z	2015-04-28T13:49:28.888027188Z	124173	ubuntu	1000	ubuntu	1000	/bin/bash	2224	/home/ubuntu/lsst/lsst_dm_stack_demo
3836857Z	2015-04-28T13:49:28.888380208Z	11638	ubuntu	1000	ubuntu	1000	/bin/bash	2224	/home/ubuntu/lsst/lsst_dm_stack_demo/bin
3875854Z	2015-04-28T13:49:28.888768616Z	10076	root	0	root	0	NaN	0	/home/ubuntu/lsst/lsst_dm_stack_demo
38080675Z	2015-04-28T13:49:29.423091724Z	11049	ubuntu	1000	ubuntu	1000	/bin/bash	2224	/home/ubuntu/lsst/lsst_dm_stack_demo/bin
33374836Z	2015-04-28T13:49:29.423381097Z	6261	ubuntu	1000	ubuntu	1000	/bin/bash	2224	/home/ubuntu/lsst/lsst_dm_stack_demo/bin
	2015-04-	16888	ubuntu	1000	ubuntu	1000	/bin/bash	2224	/home/ubuntu/lsst/lsst_dm_stack_demo/bin

Simulation

- PhD student from IHEP (Beijing, China) just started working on building a model for simulating I/O activity at the scale required by LSST, based on the observed activity

will stay at CC-IN2P3 for one year

- Simulator to be built on top of SimGrid

<http://simgrid.gforge.inria.fr>

co-supervised by Frédéric Suter, one of the main developers of SimGrid, and myself

comments & questions