Visualisation of billingDB

Sergey Kalinin

18-19 of April, 2012
dCache Workshop, Zeuthen

BERGISCHE
UNIVERSITÄT
WUPPERTAL

Tier2 at Wuppertal Description

- 1,500 cores(PBS and SGE), 1 PB storage(dCache)
- Tier3 at Dortmund also use our storage since more than one year
- Half of the top used datasets were dynamically replicated to Wuppertal
- In this talk only a dCache site considered but the others should have similar patterns

Motivation

- Sites do not have details of their storage usage but only central DDM accounting, dCache web-page and OS metrics(Ganglia, Lemon, etc).
- This tools provides the missing link and the gives a choice how the data will be presented.
- The output shown in your browser which makes it convenient to get an idea how your dCache setup works, to identify bottlenecks and problems
- The same framework can also be used for automatization

Accounting tool in brief

- Construct SQL query
- Receive an array of data(list of dictionaries)
- Convert the array to a JavaScript array
- Make plot

Details...

- Every transfer is recorded in DB called, for example, ‘billingDB’.
- DB data is read by Python modules written in Django framework. Advantages:
  - Easy to write Apps(~1 week)
  - Extremely powerful and provides you (almost) everything you need in (almost) the way you want it to have which saves your time
  - Well protected from security point of view
  - Python is perfect!
  - Consumes very little resources
- HighCharts(JS) to make awesome-looking interactive plots
- Django App is embedded in Apache/HTTP

Details:Billinginfo.model

Django makes it for you!

```
class Billinginfo(models.Model):
    datestamp =          models.DateTimeField()
    cellname =           models.CharField(max_length=200)
    action =             models.CharField(max_length=200)
    transaction =        models.CharField(max_length=200)
    pnfsid =             models.CharField(max_length=200)
    fullsize =           models.IntegerField()
    transfersize =       models.IntegerField()
    storageclass =       models.CharField(max_length=200)
    isnew =              models.BooleanField()
    client =             models.CharField(max_length=200)
    connectiontime =     models.IntegerField()
    errorcode =          models.IntegerField()
    errormessage =       models.CharField(max_length=200)
    protocol =           models.CharField(max_length=200)
    initiator =          models.CharField(max_length=200)
```

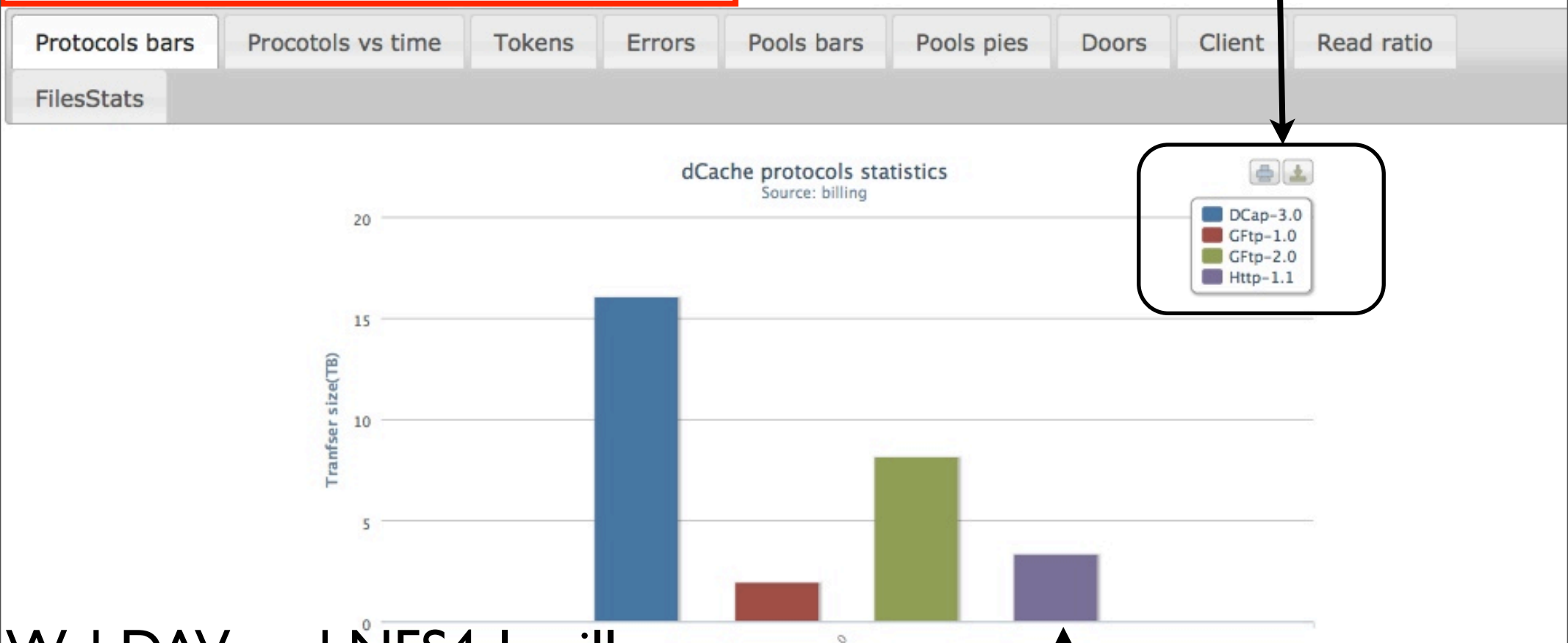
PostgreSQL Side

- By default dCache provides no indices for billingDB tables. This, of course, impact this tool as it has to go through millions of records. So, it's better to create ones if you want things to work fast. But indices are NOT necessary if you want to limit RAM/disk usage.
- The idea behind the tool is not to change much the structure of dCache databases. Only one SQL function has been added to make code elegant.
- dCache DB data are intact.

Example: protocols

Click me!
URL for the webpage in Wuppertal

The legend is clickable

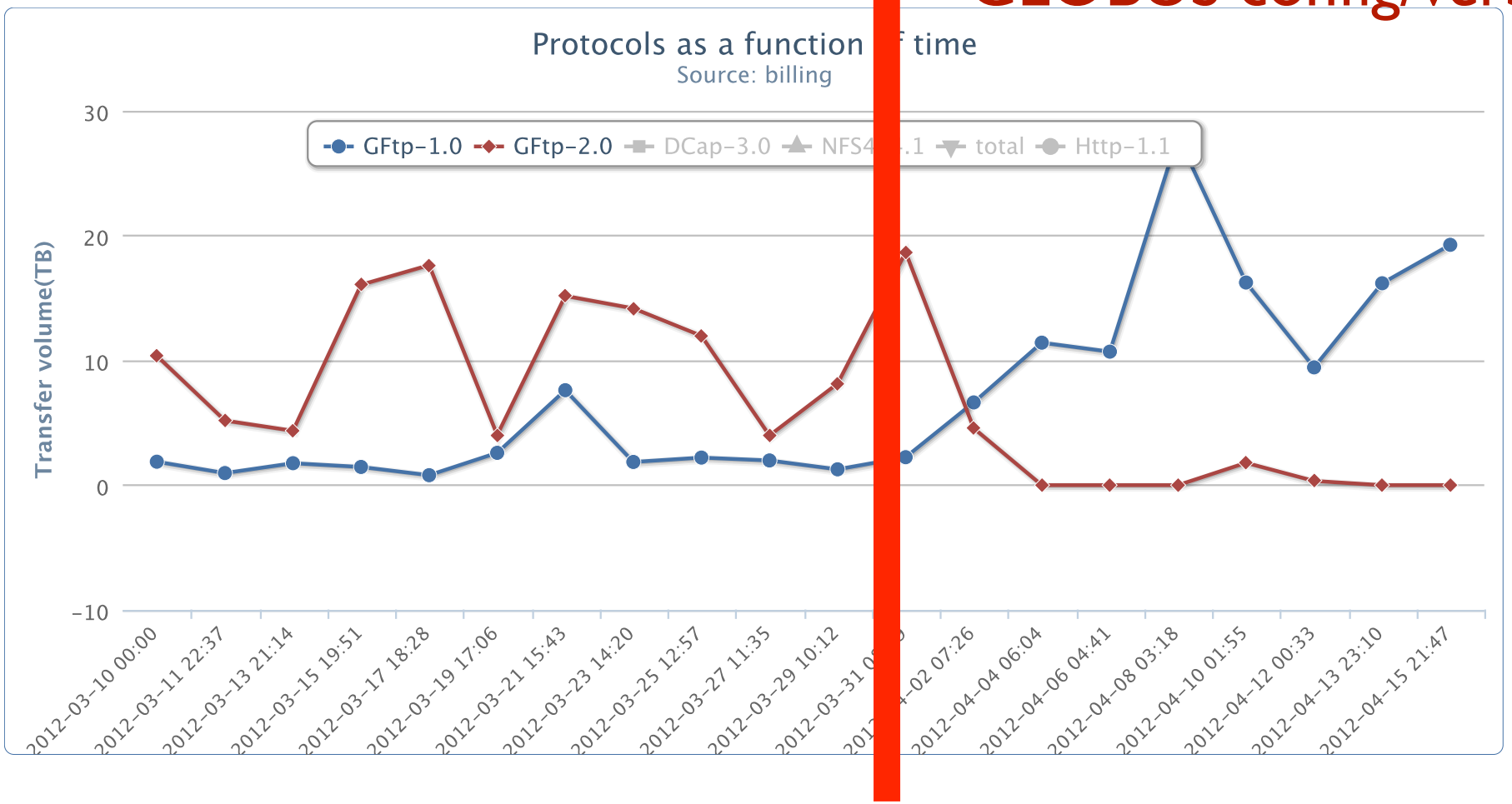


WebDAV and NFS4.1 will automagically appear here when they will be used

Http is pool2pool transfers

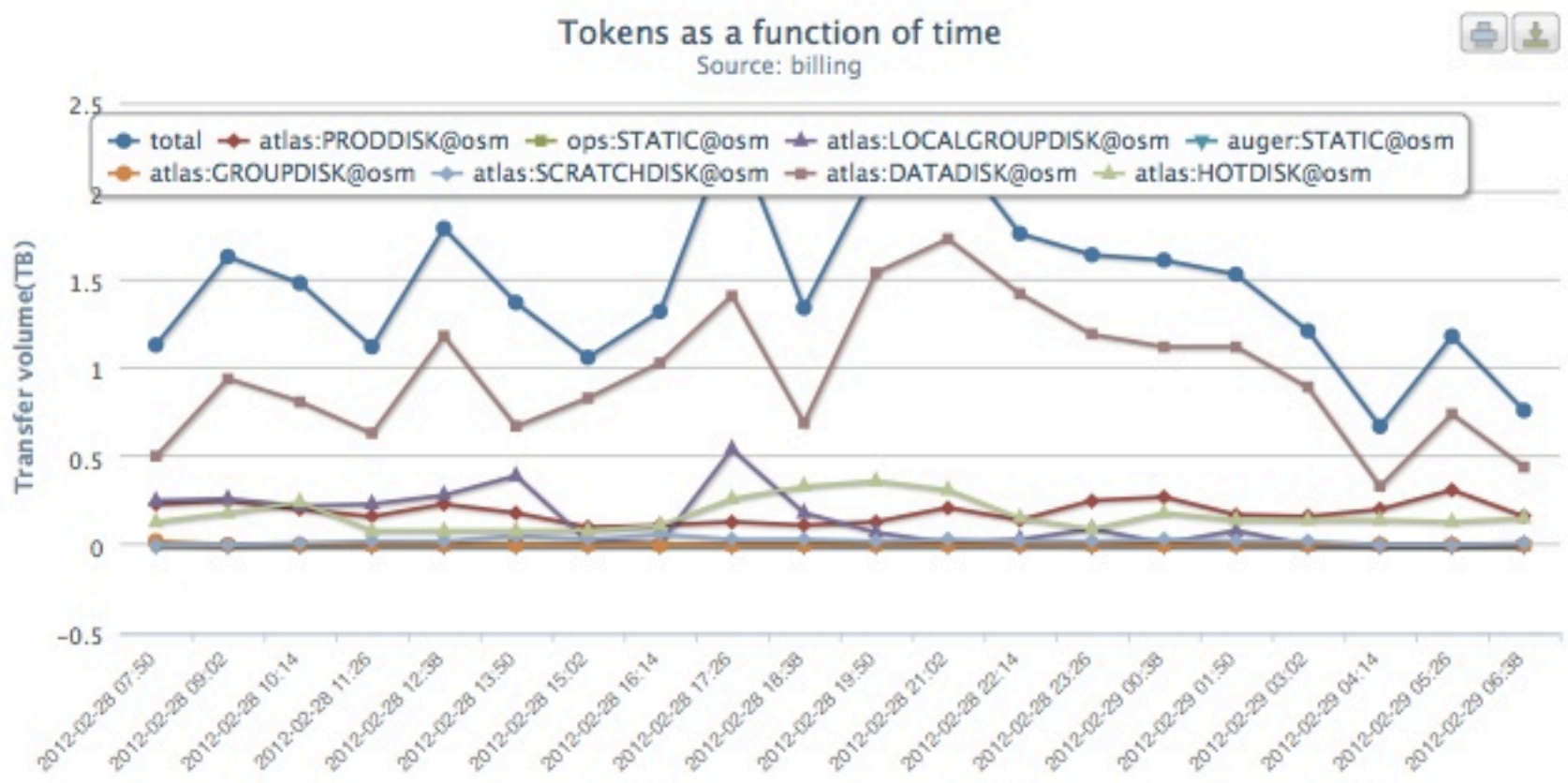
Example: protocols 2

GLOBUS config/version?



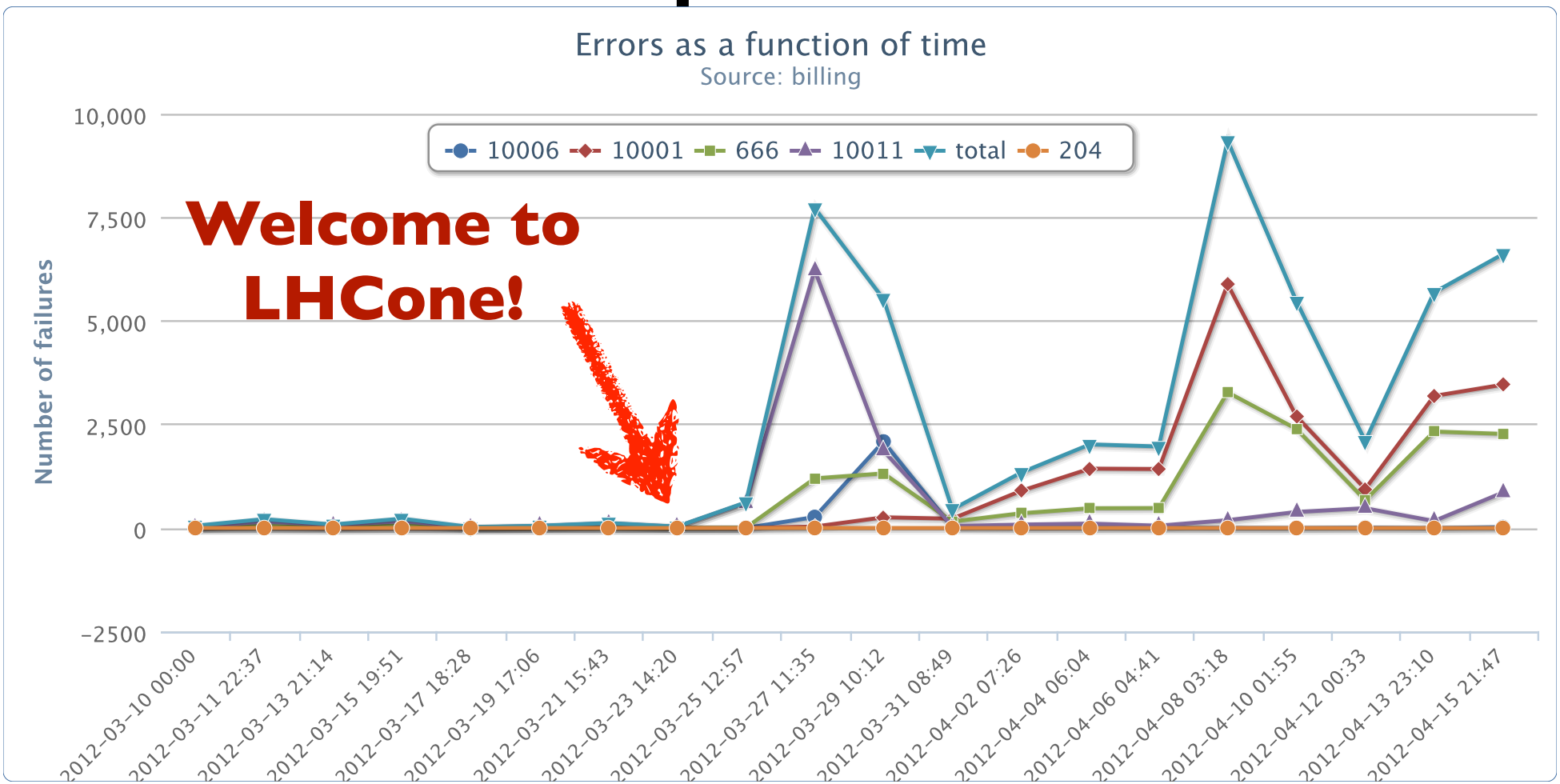
GFtp-1 goes through doors and not directly from the pools

Example: storage class



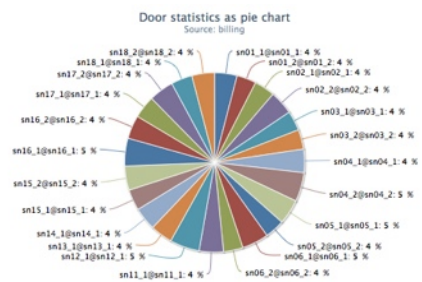
One can see very different dependencies here!

Example: errors



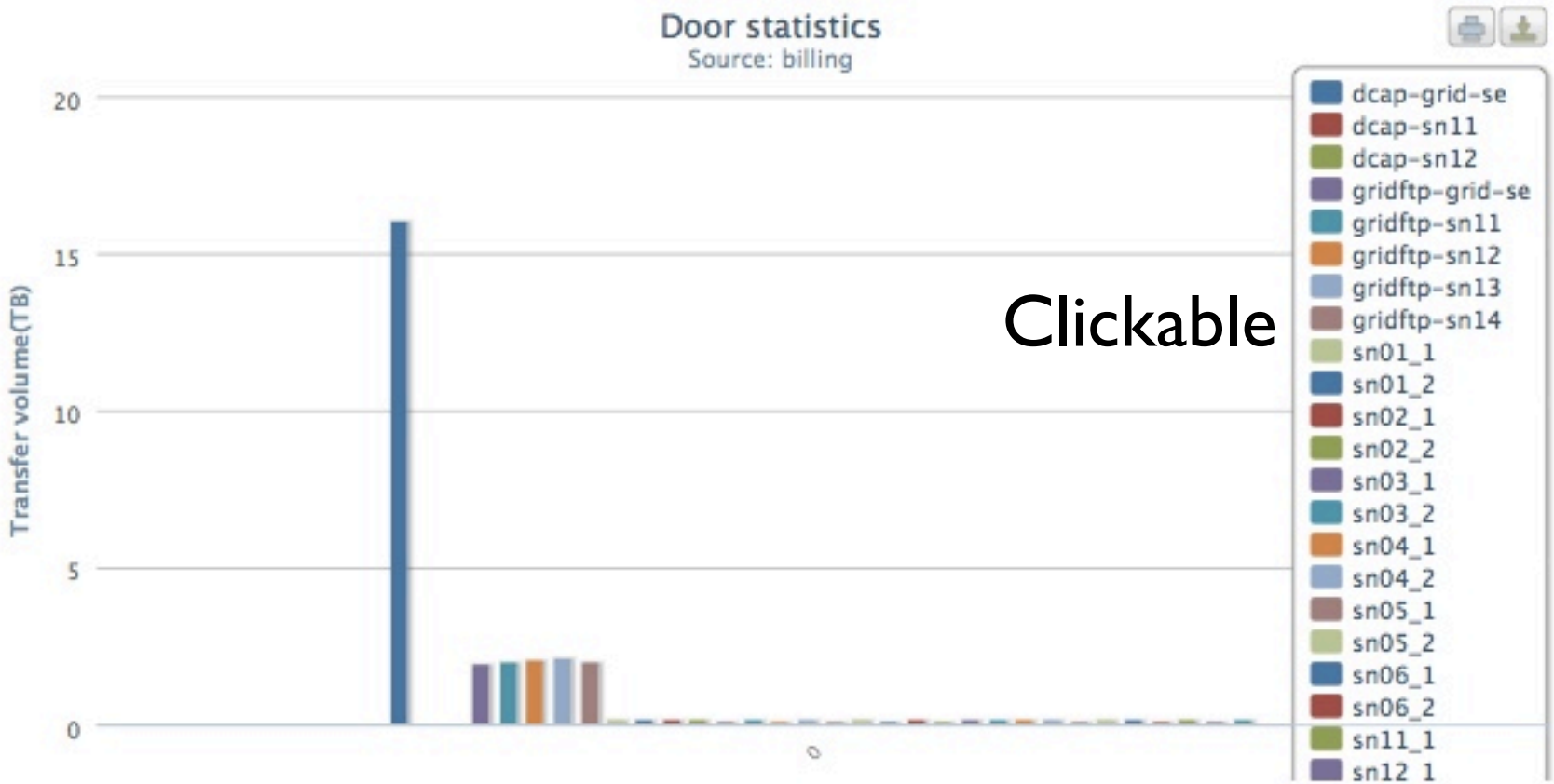
The majority of transfer error happen usually within a short time period but not always. It lets you easily see when bunches of transfer errors took place and what kind of errors.

Transfer volume per pool



We have 24 pools, each 30-40 TB in one LV. If the load balancing is perfect then this plot should look flat.

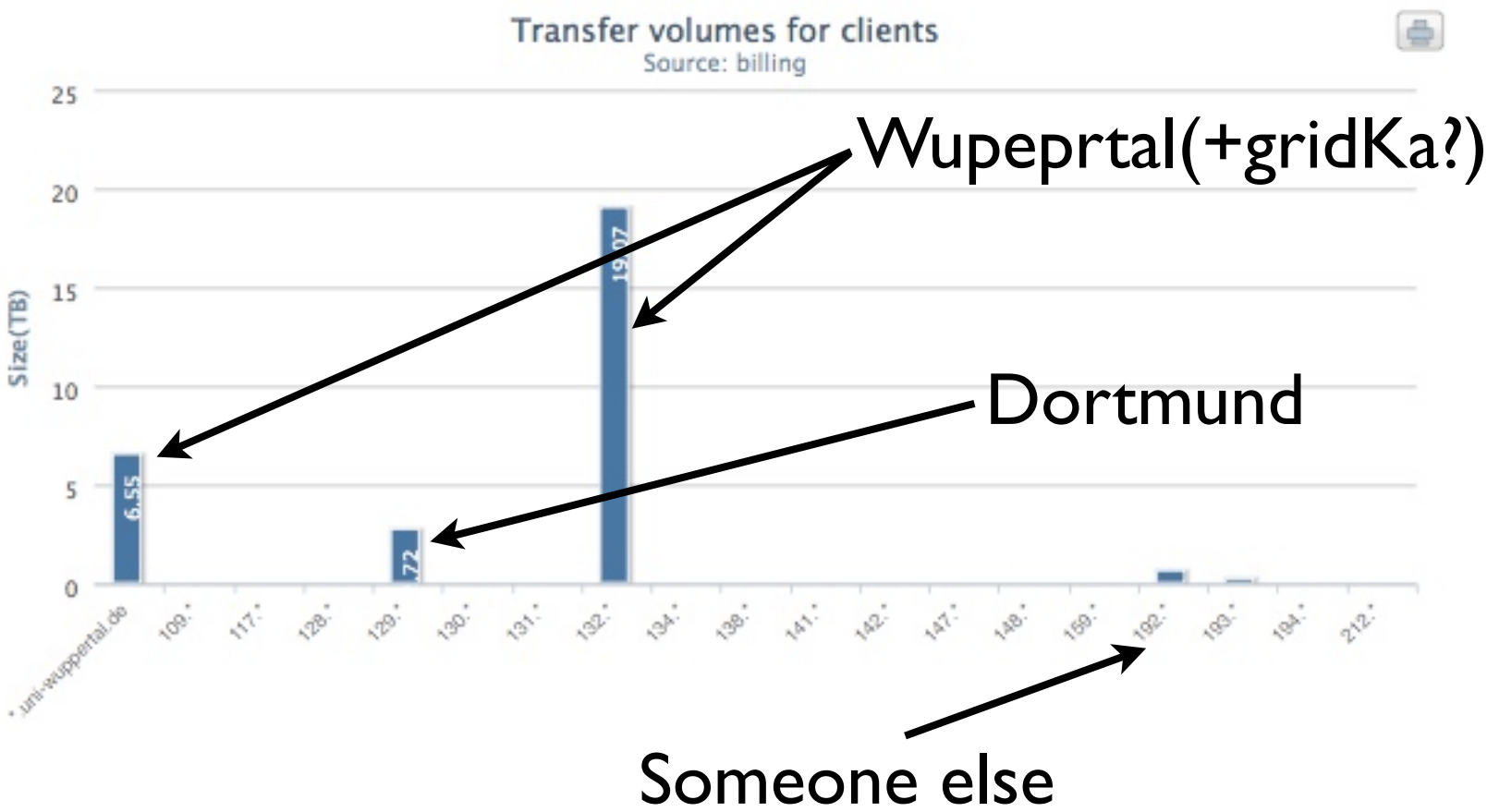
Doors



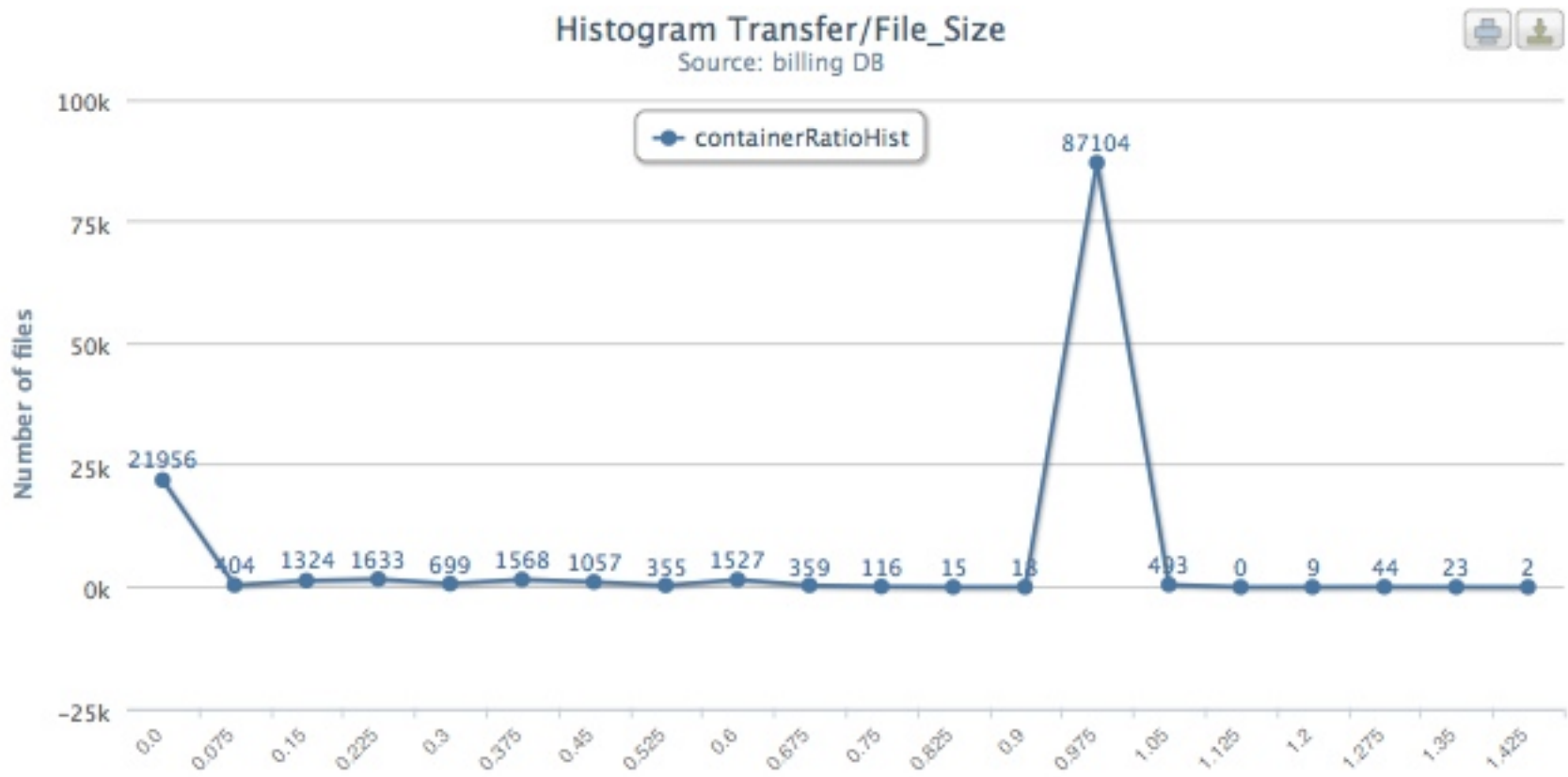
Clickable

The same thing here: it should be flat in the case of perfect load balancing. One can easily see that 2 dcap doors are not used. This picture requires proper interpretation: the traffic did not go through dcap doors while it did for gridftp doors. And http transfers are p2p.

Clients


Transfer volumes for clients
Source: billing
Size(TB)
25
20
15
10
5
0
6.55
.72
19.07
* uni-wuppertal.de
109.*
117.*
128.*
129.*
130.*
131.*
132.*
134.*
138.*
141.*
142.*
147.*
148.*
159.*
192.*
193.*
194.*
212.*
Wupeprtal(+gridKa?)
Dortmund
Someone else

Transfer Size/File Size ratio(fraction)

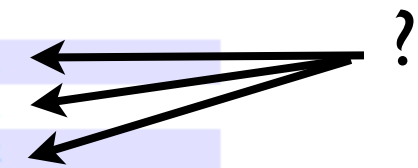


The distribution is quite broad but there are 2 visible peaks(at 0 and 1). See the next slide.

Dataset names

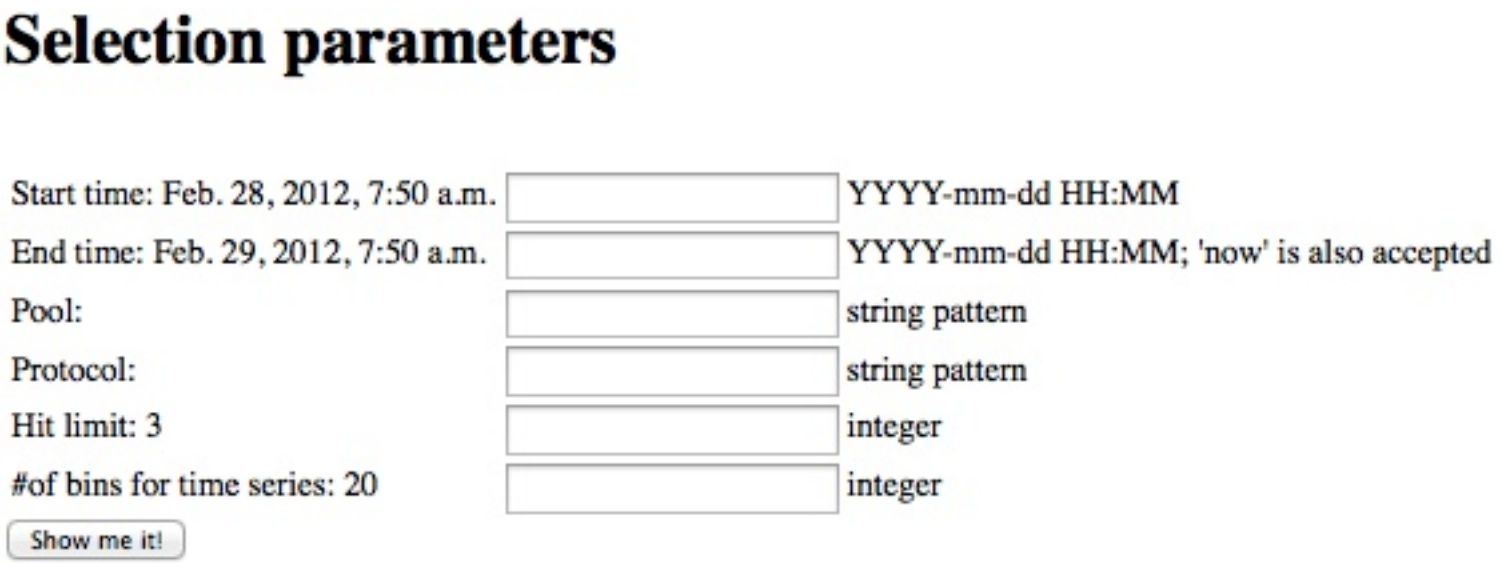
Show 50 entries Search:

Name	Size	Hits	Fraction
Deleted	3035.82	15850	0.99
ddo.000001.Atlas.Ideal.DBRelease.v16010101	1440.35	8682	1
ddo.000001.frozen.showers.DBRelease.v16080101	4621.83	8633	1
mc11_7TeV.119906.pythia_minbias_low.merge.HITS.e97...d599967_00	4970.57	8328	1
data11_7TeV.00191426.physics_Egamma.merge.NTUP_JET...d572374_00	8978.03	7700	0.09
mc11_7TeV.119907.pythia_minbias_high.merge.HITS.e9...d599969_00	814.54	5844	1
data11_7TeV.00186396.physics_Egamma.merge.NTUP_JET...d566039_00	19135.27	5155	0.1
data11_7TeV.00183462.physics_Egamma.merge.NTUP_JET...d566002_00	17648.01	4455	0.1
mc11_7TeV.107682.AlpgenJimmyWenuNp2_pt20.merge.NTU...d653547_00	2590.36	4057	0.2
mc11_7TeV.105861.TTbar_PowHeg_Pythia.merge.NTUP_SM...d653421_00	4402.38	3569	0.1
data11_7TeV.00186673.physics_Egamma.merge.AOD.f394_m934	6856.58	3374	0.61
data11_7TeV.00190933.physics_Egamma.merge.NTUP_SMW...d621766_00	2036.88	2753	0.37
data11_7TeV.00189280.physics_Egamma.merge.NTUP_JET...d564682_00	3246.91	2690	0.09
data11_7TeV.00189483.physics_Egamma.merge.NTUP_JET...d564674_00	3281.77	2630	0.08
data11_7TeV.00183462.physics_Egamma.merge.AOD.r260...d493619_00	8609.56	2601	0.76
data11_7TeV.00183581.physics_Egamma.merge.NTUP_JET...d566005_00	9639.69	2567	0.1
data11_7TeV.00191190.physics_Muons.merge.NTUP_SUSY...d607920_00	985.09	2564	0.34
data11_7TeV.00186923.physics_JetTauEtmiss.merge.AO...d513593_00	8882.32	2445	0.23
user.sangelid.NEW1.mc11_7TeV.105200.T1_McAtNlo_Jim...b053705599	0	2115	1
data11_7TeV.00189822.physics_Egamma.merge.NTUP_TOP...d602036_00	1510.38	2100	0.21
data11_7TeV.00183286.physics_Egamma.merge.AOD.r260...d493625_00	7024.74	2083	0.94
data11_7TeV.00183544.physics_Egamma.merge.NTUP_JET...d566003_00	8178.97	2061	0.11



NTUPs are the majority of transfers and they use ~10% per job. Usage of AODs is pretty efficient as well as HITS which are inputs. The field 'Search' allows you to filter the names and it comes for free for jQuery tables.

Configurability of selection



There is a number of selection parameters available:

- Time
- Pool names(e.g. 'sn0' in the case of Wuppertal)
- Protocol
- Storage classes, etc.

The operator is 'AND' between parameters

The case of Dortmund

T3 in Dortmund has significant CPU power but no storage. dccp/lcg-cp from/to Wuppertal. So, filtering on DO IPs:

Name	Size	Hits	Fraction
ddo.000001.frozen.showers.DBRelease.v16080101	371.55	694	1
ddo.000001.Atlas.Ideal.DBRelease.v16010101	115.14	694	1
Deleted	23.94	628	1
ddo.000001.Atlas.Ideal.DBRelease.v180501	17.82	30	1
panda.735660.03.09.EVNT.2f853de4-119a-402b-84d0-f93abad6cfdc	2.9	48	1
mc10_7TeV.132655.simplifiedModel_12_1200_0.evgen.E...d332162_00	1.36	8	1
panda.733349.03.08.EVNT.1b8b2dce-fc72-4bba-a0f1-5c9e50652a89	1.27	14	1
panda.735660.03.09.EVNT.19aa56c5-361a-4603-9abc-f56801ec14e9	1.27	21	1
mc10_7TeV.105200.T1_McAtNlo_Jimmy.evgen.EVNT.e844_...d456778_00	1.13	7	1
mc10_7TeV.124168.MSSM_s1600_g2000_x00_herwigpp_sus...d303225_00	1.04	7	1
mc10_7TeV.132634.simplifiedModel_12_900_150.evgen....d332141_00	1.02	6	1
mc10_7TeV.117560.toprex_tt_bWqgamma_mt170.evgen.EV...d208854_00	1.01	15	1

One easily sees that installing a Squid and usage of curl would decrease the traffic between the sites.

Things which could be made with several lines of code

- Trigger replication hot files or hot datasets. Or in general introduction of any kind of trigger actions is easy.
- Make your own plots(e.g. GridFTP deletion rate).
- Live data feed is also possible(see HighCharts demos)
- One can demonize it to perform automatic procedures(e.g. send an e-mail in case of growing failure rate).
- One can integrate it into any monitoring pages or use on its own.Completely independent from dCache but can be integrated
- You can do something very exotic. HighCharts plots are vector graphics and you can use them, for example, as icons for HappyFaces tabs.
- Use cookies for user preferences

Conclusions

- A powerful and (yet another) monitoring tool is developed at the University of Wuppertal.
- Highly customizable and provides what dCache admins would like to see in the fashion they like it.
- I can only advertise Django and HighCharts for those who deal with databases and its web representations.
- accounting.tgz