File Catalog tutorial and distributed disk usage

Introduction ...

The people : Nikita Soldatov, Adam Kisiel, myself, …

Why do we need a FileCatalog ??

- Number of files in STAR is ~ 2 M (will get worst, far worst …)

- Information structure complex ...
  - production, library
  - filetype, size, geometry
  - collision, magnetic field, trigger setup name
  - ... but we (are supposed to) keep information about triggers and counters, finding a data-set requires strong Cataloguing

API

One existing complete user API (written in perl), some C
a command line interface

```
% get_file_list.pl
```

How do I use it ??

- Getting a quick help reminder

```
% get_file_list.pl

... bla bla bla ... some help that is ...

  all available bbc collision configuration createtime
datetaken eemc emc events extension filecomment filename
fileseq filetype fpd ftpc fulld fulls gencomment generator
genparams genversion geometry inserttime lgnm lgpth
library limit magscale magvalue md5sum node noround
nounique owner path persistent pmd prodcomment production
protection rich runcomments runnumber runtype sanity
simcomment simulation site sitecmt siteloc size ssd
startrecord storage stream svt tof tpc trgcount
trgdefinition trgname trgsetupname trgversion trgword
```

- Documentation is available at
  /STAR/comp/sofi/FileCatalog/

Syntax

General syntax (“{“ indicates optional list “}”)

```
% get_file_list.pl {-qualifier} -keys 'key1{,key2,…}'
-cond 'key1 op1 value{,key2 op2 value2,…}'
```

```
% get_file_list.pl -keys path,filename -cond storage=NFS

/star/data24/reco/UPCCombined/FullField/P03ia/2003/074::st
_physics_4074004_raw_0040013.MuDst.root
```

Returned values are separated by “::” by default

Use `-delim '/'` for example to have ‘path/filename’ automatically

```
% get_file_list.pl -keys storage -cond
filename=rcf0183_02_300evts.geant.root
```

returned values requested with `-keys` are interchangeable with conditions in `-cond` ; `-cond` however requires a value and operator restriction (modulo the one displayed in italic in the preceding slide)

Possible Operators

<=	Not greater than	
<	Lesser than	
>=	Not less than	
>	Greater than	
<>	Not equal to	
=	equal to	
!~	Not containing (i.e. do not match)	strings
~	Containing (i.e. approximately matching)	strings
[]	In range	
][	Outside the range	
%	Modulo	integer
%%	Not Modulo	integer

Welcome to the World of replica Catalogs.

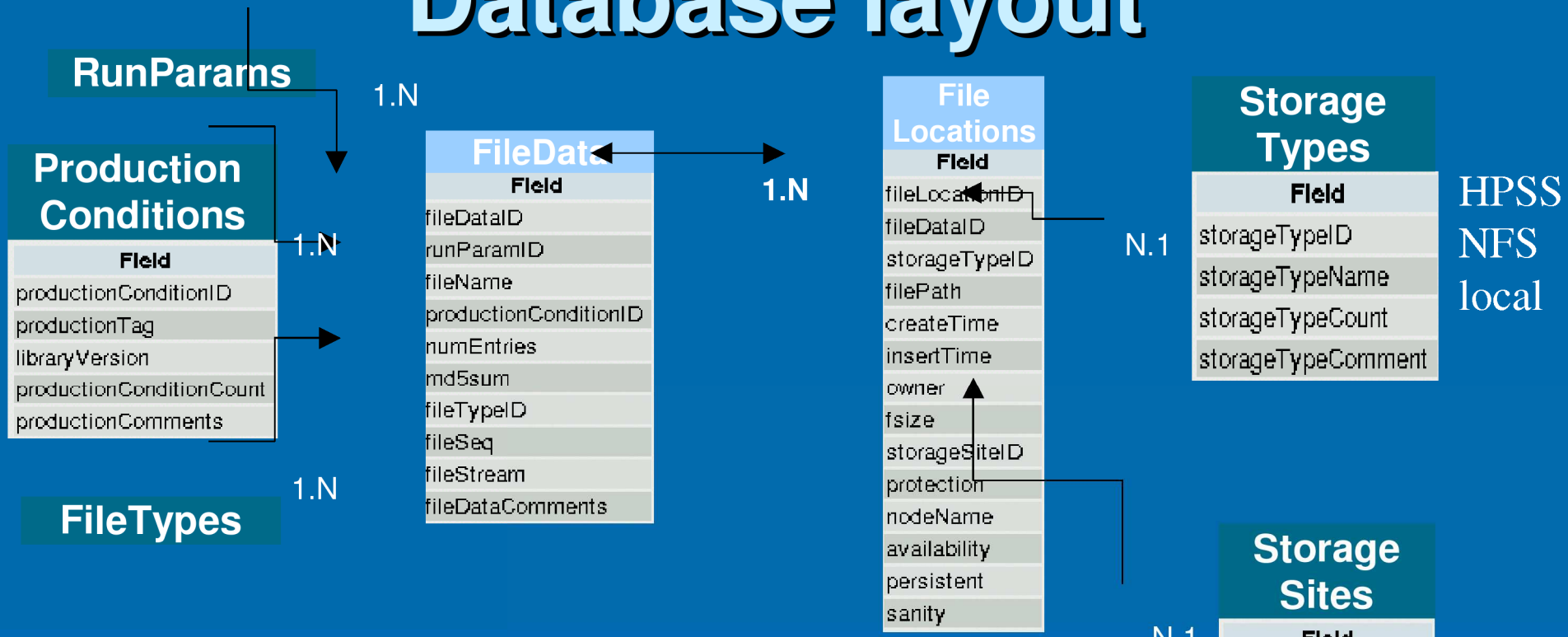
- Number of files in STAR ~ 2 M
  - That's a lie !!! Total = 3 M with replicas : File have more than one location

  site Be aware of site=BNL, site=LBL
  node 'localhost' by default
  storage NFS, local, HPSS
  path itself within a 'storage'

  unconstraint, path and filename are NOT unique key pairs
  (use `-distinct` to ensure it ; `-onefile` ensures one instance of a file)

  Number of files on centralized storage : 617986
  NFS, disk visible from anywhere in the facility (path ~ /star/dataXX)
  Number of files on local disk : 131886
  local disk are visible only from a unique node

Database layout



Site, node, storage and path forms the unique key for FileLocations

`/tmp/bla.root` **cannot be unique**

```
BNL somenode.domain NFS /tmp/bla.root IS
```

Meta Data

Locations / Replicas

Typical Examples

- How to locate files within a specific trigger setup ??

```
% get_file_list.pl -keys path,filename -cond
trgsetupname=UPCCombined
```

will lead to a long (100 records) list of possible files with path

```
% get_file_list.pl -keys storage -cond
trgsetupname=UPCCombined
```

this will give you all possible storage type for the trigger setup name UPCCombined

In general, for listing all possible values for a keyword, use

% get_file_list.pl -keys keyword –distinct {-alls}

```
% get_file_list.pl -keys path,filename -cond
trgsetupname=UPCCombined,storage=NFS,
filetype=daq_reco_MuDst
```

Typical Examples

- But but … I always get only 100 records

  That's normal, it is the default. Use –limit to change the number of records, full list with –limit 0.

- A few handy querries

  I know a simulation file name, how do I get the geometry configuration ?

```
% get_file_list.pl -keys geometry -cond
filename=rcf0183_02_300evts.geant.root -distinct
Year2001
```

  Which production and geometry ?

```
% get_file_list.pl -keys production,geometry -cond
filename=rcf0183_02_300evts.geant.root -distinct
P01gl::year2001
P01gk::year2001
P02gb::year2001
```

Aggregate Operation

- Can also do queries leading to summary information

```
% get_file_list.pl -keys
'sum(sanity),sum(size),sum(events),grp(trgsetupname)'
-cond collision=auau200,sanity=1,production=P02gc
173528::71128970908::2174::central
2194995::754986154611::20313::productionCentral
635075::372522928644::11280::productionCentral1200
4635741::1663580227269::53992::productionCentral600
8808076::1011162248161::40914::ProductionMinBias
```

One more concept & future

- The keyword sanity is used for two case
  - The file is corrupted (ROOT IO will crash your application)
  - The file is NOT good for Physics
  - You MUST use sanity=1 to get the good files

- Future (not yet available)

```
% get_file_list.pl -keys path,filename -cond
trgname=ppBHT1-fast&&ppFPDW-fast,sanity=1
```

  already “in place”, only need to fill the database consistently (not done this year)

```
% get_file_list.pl -keys path,filename -cond
tpcOK=1,ftpcOK=1,sanity=1,…
```

  Not implemented, we plan to add a detector readiness flag

Distributed disk ??

- Shall I sort this manually ??
  - You can always ask for
    ```
    % get_file_list.pl -cond node,path,filename -cond
    storage=local,sanity=1,…
    ```
    and dispatch by hand ut why ??

- The Scheduler
  - Does this for you (examples in next talk) : `fileListSyntax`, `preferStorage`
  - There is NO need to use `-distinct` or `-onefile`

- Notes
  - Yes, please, use the sanity flag …
  - Use the Scheduler (it is a key component of our Grid approach)
  - Any Scheduler `URL="catalog:star.bnl.gov?...` can (and should) be checked from the command line using get_file_list.pl . If it does not work from the command line, it is NOT a Scheduler problem.