Index files for Belle II - very small skim containers

Tristan Bloomfield, Chia-Ling Hsu, Martin Sevior,
University of Melbourne
Thomas Kuhr,
Ludwig-Maximillians-Univeristät München (LMU)
I. Ueda, H. Miyake, T. Hara
KEK
For the Belle II collaboration

Data analysis strategy

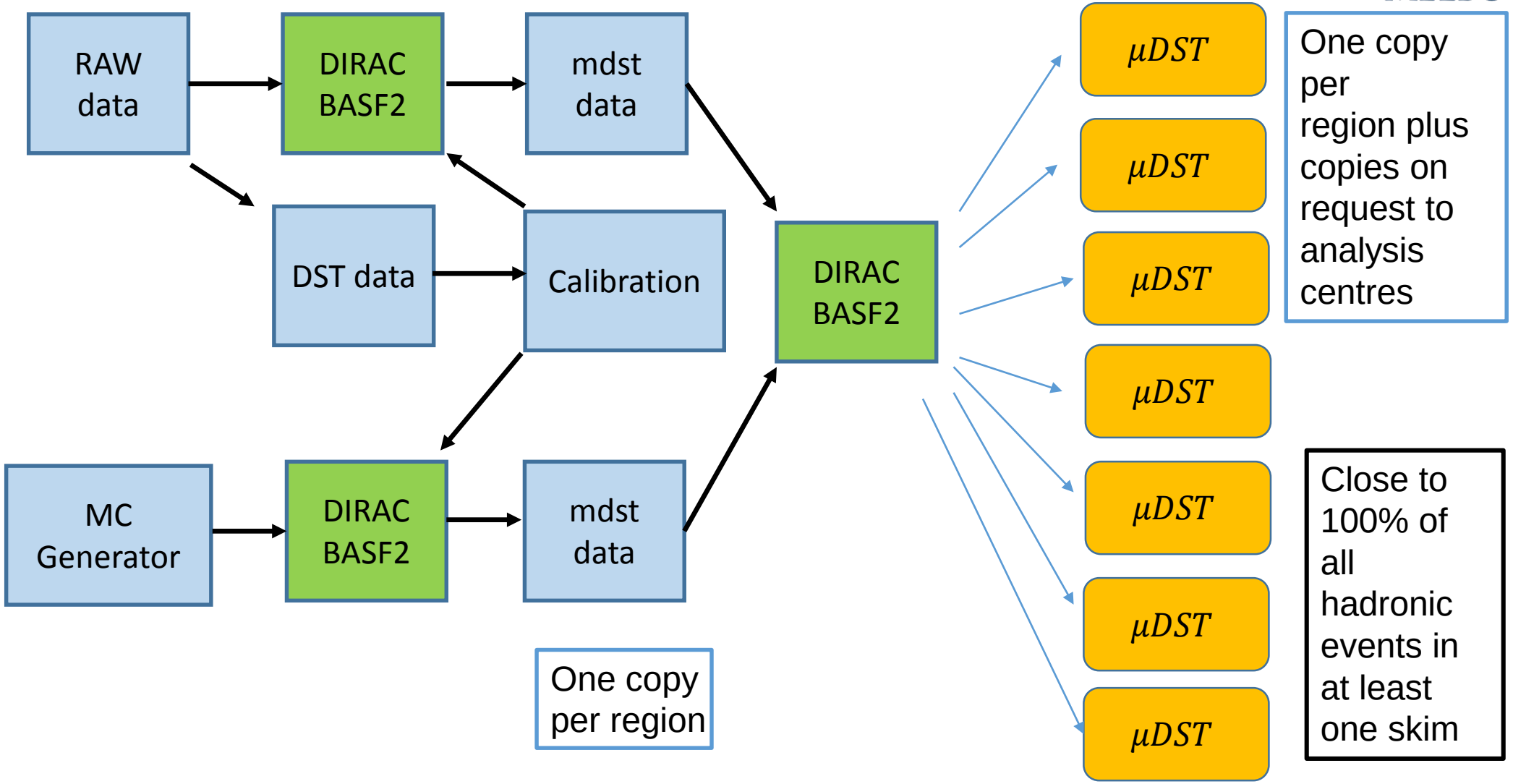
Centrally managed

- Raw Data => DST=> Calibrate Data
- RAW DST => mini DST (mDST)
- Skim mDST to make various datasets enriched in Physics events (eg $J/\psi, \tau$ −pair, 2-body charmless etc.)

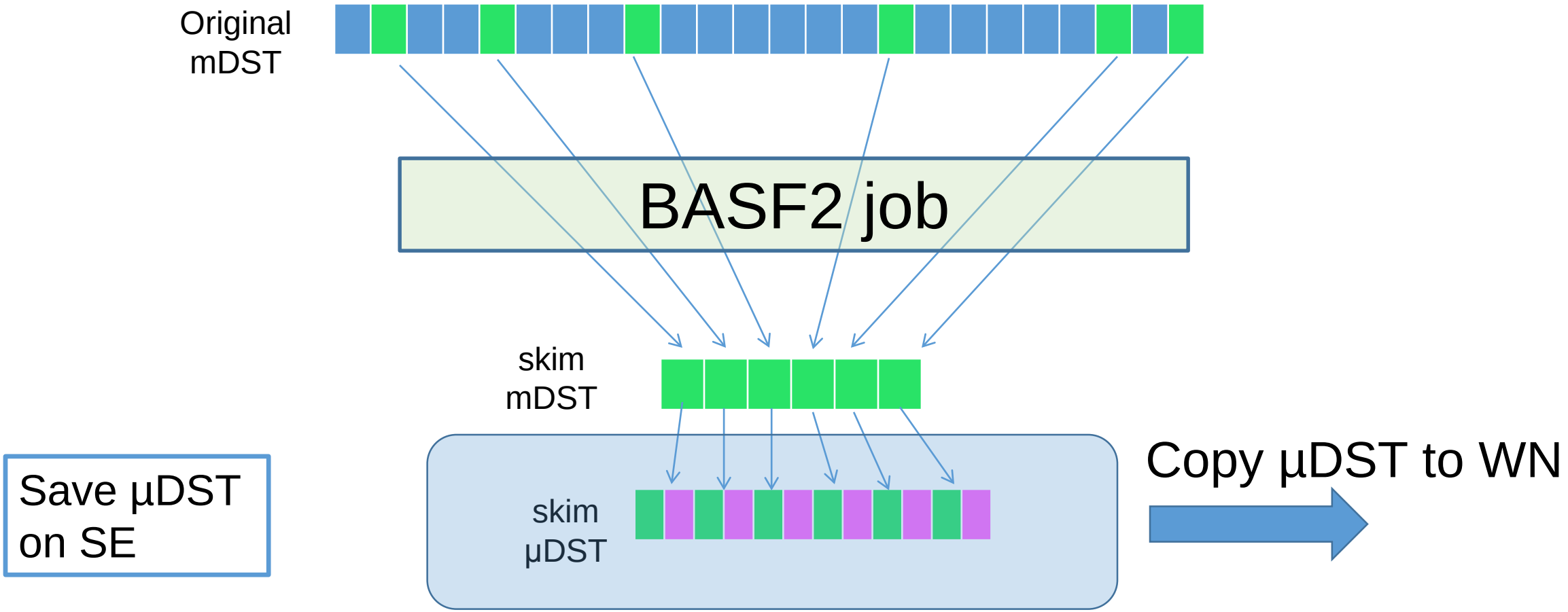
Individual Physicists

- Analyze skim datasets on grid to make root-ntuples
- Analyze ntuples on local resources (workstations, local clusters)

Data flow for Skimming

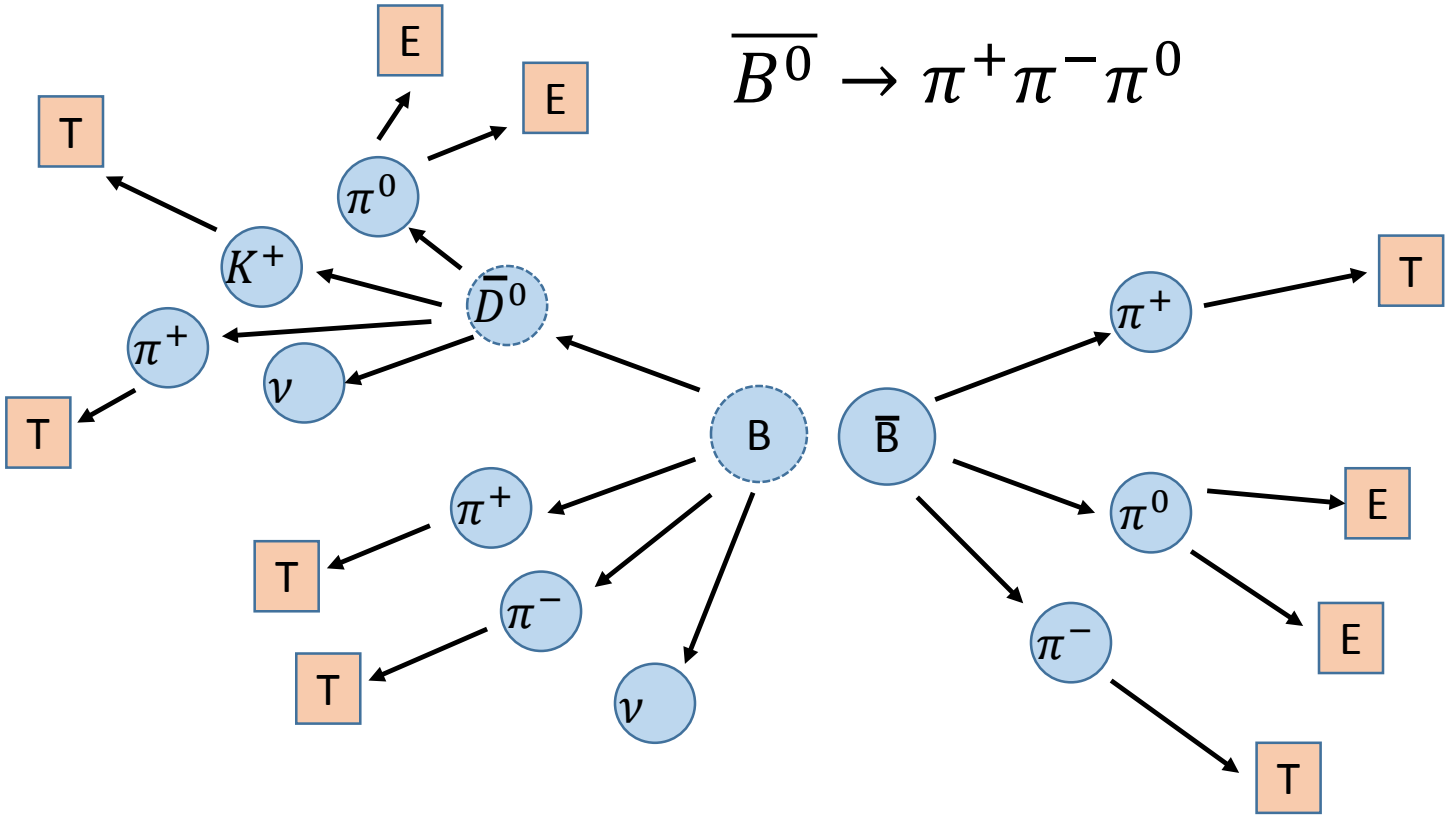


Skim Files – copy µDST to WN



mDST/μDST

$$\overline{B^0} \to \pi^+\pi^-\pi^0$$

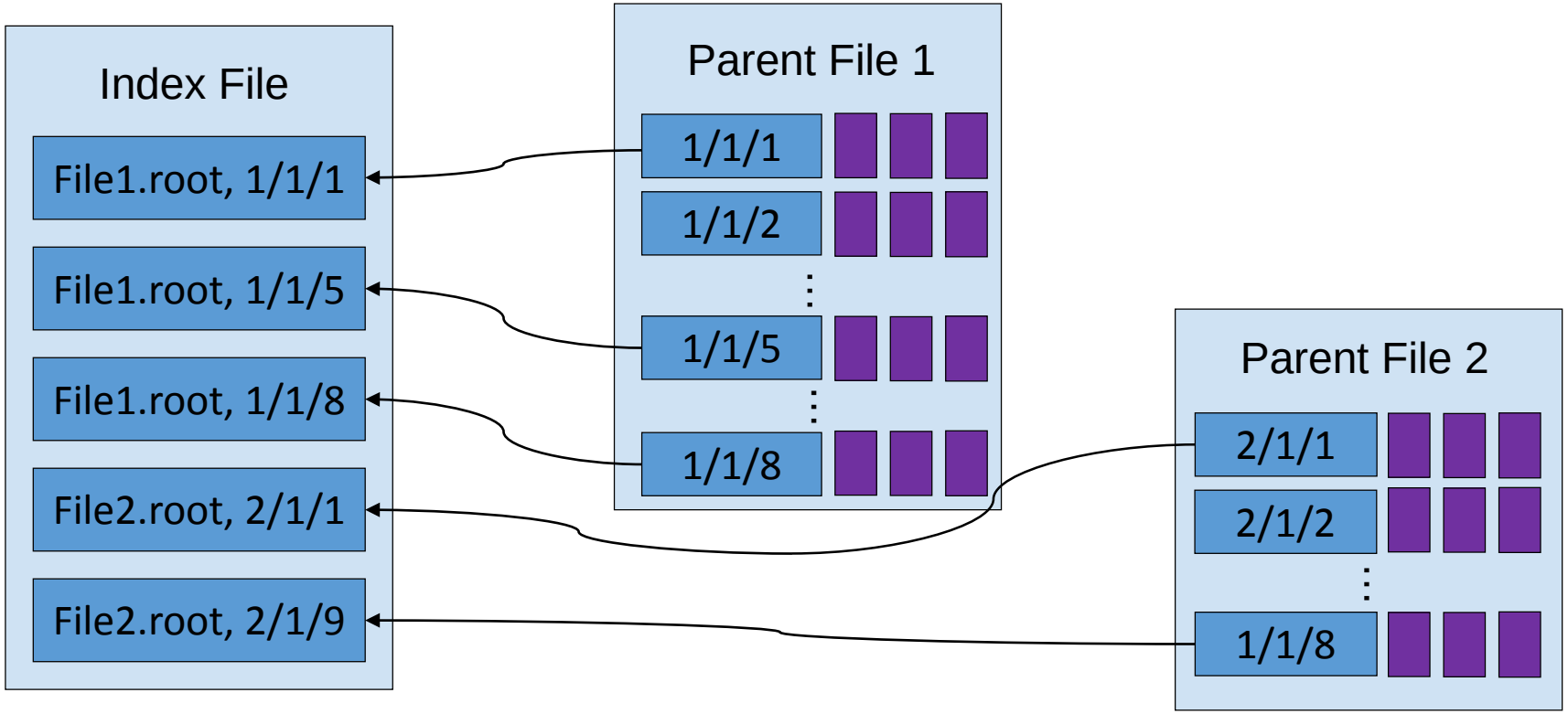


mDST, Keep: T E E T T T E T T

μDST, Keep: T E E T T T E T T π^- π^0 π^+ $\bar{B}$ π^- π^+ K^+ π^0 π^+

Index skims

- Instead of writing out actual data after a skim, we simply record the parent file and the event number that passed the skim
- Use xrootd protocol to retrieve data



Size of skim formats

- File-size of mDST/µDST/index formats depends on skim modes
- For $B \rightarrow hhh$ modes index typically ~ **0.3% size of mDST**

	Number of Events (Skimed / Unskimmed)	Unskimmed MDST size (KB)	MDST skim size (KB)	MDST size / event (KB)	µDST skim size (KB)	µDST size / event (KB)	µDST skim size (% of MDST)	Index skim size (KB)	Index size / event (KB)	Index skim size (% of MDST)
B0B0bar	176 / 50000	341,455.10	1063.98	6.05	1217.04	6.92	114.39%	16.09	0.0914	1.51%
B-B+	633 / 90000	607,952.66	3583.03	5.66	4057.14	6.41	113.23%	21.10	0.0333	0.59%
ccbar	4667 / 80000	444,892.34	24,721.68	5.30	28,196.32	6.04	114.06%	60.51	0.0130	0.24%
ddbar	6643 / 90000	442,686.86	32,820.92	4.94	37,816.15	5.69	115.22%	79.69	0.0120	0.24%
ssbar	7672 / 90000	459,654.17	39,049.41	5.09	44,882.33	5.85	114.94%	89.43	0.0117	0.23%
uubar	7950 / 90000	443,066.46	39,095.37	4.92	45,054.99	5.67	115.24%	92.07	0.0116	0.24%

Definitions

- Logical File Name (LFN): In BelleII LFN is the subpath to a file from the top BelleII directory on a site. This will be the same for all replicas of a file on all sites.
- Physical File Name (PFN): This is the actually address used to access the file. XRD URL for a file streamed via XRootD.
- Logical File Catalogue (LFC): Maps an LFN to PFNs (one for each site hosting a replica).

LFN

belle/user/tbloom/index_test_rho_gam/
test_mdst_000002_prod00000050_task00000002.root

PFNs (Replicas on Melbourne and Napoli via XRD)

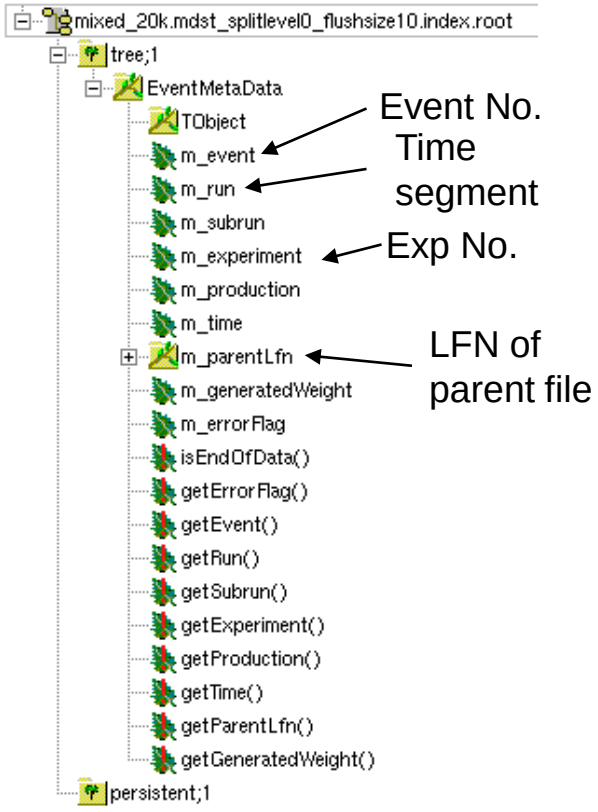
root://b2se.mel.coepp.org.au:1094//dpm/mel.coepp.org.au/home/belle/TMP/belle/user/tbloom/index_test_rho_gam/test_mdst_000002_prod00000050_task00000002.root

root://belle-dpm-01.na.infn.it//dpm/na.infn.it/home/belle/TMP/belle/user/tbloom/index_test_rho_gam/test_mdst_000002_prod00000050_task00000002.root

Reading Index Files in BASF2

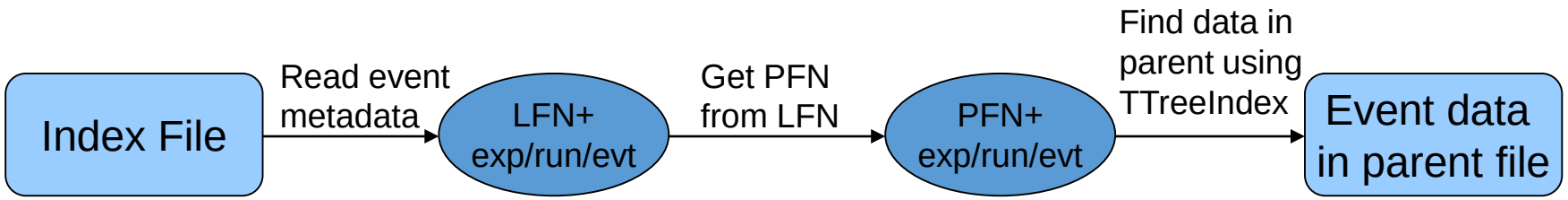
- When basf2 outputs a root file, metadata is written for each event with the Parent LFN string and exp/run/event numbers.
- When reading an index file, Basf2 reads metadata for every event in the index file and makes list of parent file LFNs.
- Once basf2 has the list of parent LFNs, it uses these to find the corresponding PFN.

Event Metadata

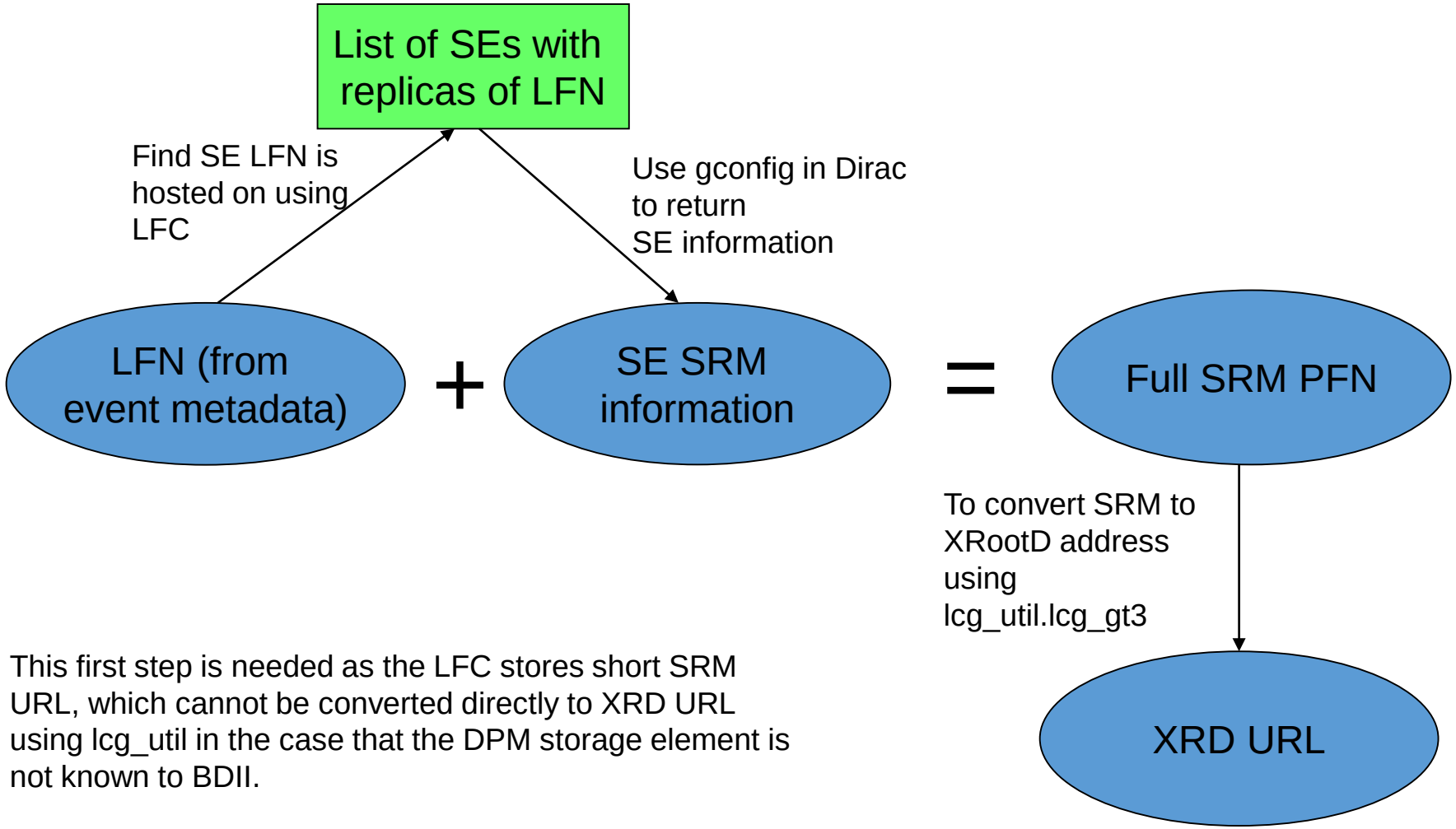


Reading Index Files in BASF2

- Basf2 uses this PFN to access parent files using standard ROOT I/O.
- basf2 opens the event data by using the TTreeIndex in the parent file, which uniquely maps the exp/run/evt to root file event.



Finding XRD URL from LFN



This first step is needed as the LFC stores short SRM URL, which cannot be converted directly to XRD URL using lcg_util in the case that the DPM storage element is not known to BDII.

Index file Performance vs Root parameters

"Random" access of events depends on root file format parameters.

- Range of parameters to test splitlevel = (0,1,2,5,10) and autoflushsize = (10,100,1000).

Index Read Speed (Pass rate 0.035%)

- mDST with 200,000 generic B MC5 events skimmed to 685 events (0.35% pass rate)
- TTreeCache was disabled.
- Average of 300 runs at KEKCC.
- Default (splitLevel=99, autoflushsize=-30000000): 33.96±1.77ms

autoflushsize/ splitlevel	**10**	**100**	**1000**
0	18.77 ±2.70ms (55±8%)	34.08±3.82ms (100±11%)	35.38±1.84ms (104±5%)
1	28.23±3.97ms (83±12%)	30.48±3.10ms (90±9%)	59.24±5.67ms (174±17%)
2	25.29±3.43ms (74±10%)	30.85±3.19ms (91±9%)	54.34±4.94ms (160±15%)
5	22.51±2.90ms (66±9%)	36.43±2.88ms (107±8%)	57.40±4.30ms (169±13%)
99	29.04±2.57ms (86±8%)	39.26±3.48ms (116±10%)	56.06±4.68ms (165±14%)

Index Read Speed (Pass Rate = 5.8%)

- mDST with 200,000 generic B MC5 events skimmed to 11553 events (5.78% pass rate).
- TTreeCache was disabled.
- Average of 76 or 77 runs at KEKCC.
- Default (splitLevel=99, autoflushsize=-30000000): 19.98±0.38ms

autoflushsize/ splitlevel	**10**	**100**	**1000**
0	21.39±0.51ms (107±3%)	22.45±0.66ms (112±3%)	20.28±0.28ms (101±1%)
1	20.89±0.30ms (105±2%)	22.26±0.54ms (111±3%)	22.22±0.42ms (111±2%)
2	20.79±0.21ms (104±1%)	21.97±0.35ms (110±2%)	21.86±0.36ms (109±2%)
5	20.77±0.18ms (104±1%)	22.36±0.29ms (112±1%)	21.99±0.39ms (110±2%)
99	20.93±0.18ms (105±1%)	22.44±0.34ms (112±2%)	21.97±0.34ms (110±2%)

mDST Skim size

- Size of mDST file with new ROOT parameters compared to original mDST (67.21MiB).
- 100 runs, all runs with same parameters returned same file size (as expected).

autoflushsize/ splitlevel	**10**	**100**	**1000**
0	75.60MiB (112%)	70.90MiB (105%)	70.39MiB (105%)
1	106.79MiB (159%)	71.63MiB (107%)	67.97MiB (101%)
2	116.11MiB (173%)	71.08MiB (106%)	67.24MiB (100%)
5	116.11MiB (173%)	71.08MiB (106%)	67.24MiB (100%)
99	116.11MiB (173%)	71.08MiB (106%)	67.24MiB (100%)

Index vs mDST

- Impact of Index/mDST read speed on full basf2 run times for estimated $B \rightarrow h^{\pm}h^{\mp}h^{\pm}$ and full reconstruction.

	mDST Default	**mDST** (Flush=10, split=0)	**Index** Default	**Index** (Flush=10, split=0)
RootInput time only				
0.35% pass	0.18ms (100%)	0.25ms (138%)	33.96ms (18866%)	18.77ms (10427%)
5.78% pass	0.18ms (100%)	0.25ms (138%)	19.98ms (11100%)	21.39ms (11883%)
Full run time ($B \rightarrow h^{\pm}h^{\mp}h^{\pm}$)				
0.35% pass	2.06ms (100%)	2.13ms (103%)	35.84ms (1740%)	20.65 (1002%)
5.78% pass	2.06ms (100%)	2.13ms (103%)	21.86ms (1061%)	23.27ms (1130%)
Full run time (Full Reconstruction)				
0.35% pass	60.38ms (100%)	60.45ms (100%)	94.16ms (156%)	78.97ms (131%)
5.78% pass	60.38ms (100%)	60.45ms (100%)	80.18ms (133%)	82.59ms (137%)

Order of magnitude increase in analysis time per event of index vs mDST

Summary

- Substantial reduction in file size using index files.
- mDST/µDST skims use substantially less CPU.
- Root optimization found improvement in index performance for low pass rate skims however not for higher pass rate, also worse mDST performance.
- Only autoflushsize of 10 offers improved index performance.
- Best setting gives 82% increase in index read speed, but 28% (37%) decrease in mDST read (write) speed.
- Implemented a method to get a XRD URL from PFN using DIRAC and LFC.

Backup

Getting PFN from LFN

- A PFN needs to be found from the LFN of all the parent files of an index file.
- LFC currently only stores PFNs as short SRM URLs, to use another protocol PFN needs to be generated from LFN at runtime by gbasf2.
- LFN -> PFN currently implemented for XRootD.
- Constructs the full SRM URL of the parent file, then uses lcg_util to convert to an XRD URL.

Backup: BASF2 and PFNs

- Basf2 does not know about the GRID (including the LFC), it gets the PFN for an LFN by a simple lookup of a temporary local file listing LFNs and corresponding PFNs.
- This temporary local file needs to be generated by gbasf2 before basf2 runs by combining data from the LFC and DIRAC (as shown on slide 16).
- The file is then attached to the job sandbox and read when basf2 runs on the grid.

Compression Test

- Using gzip with compression of -9
- Default: 1331.38MiB -> 1318.85Mib (99.06%)
- Split=0,Flush=10: 1511.86 -> 1487.96 (98.42%)

mDST Read Speed

- Speed of reading new mDST normally in basf2
- 100 runs at KEKCC.
- Default (splitLevel=99, autoflushsize=-30000000): 0.18±0.00ms

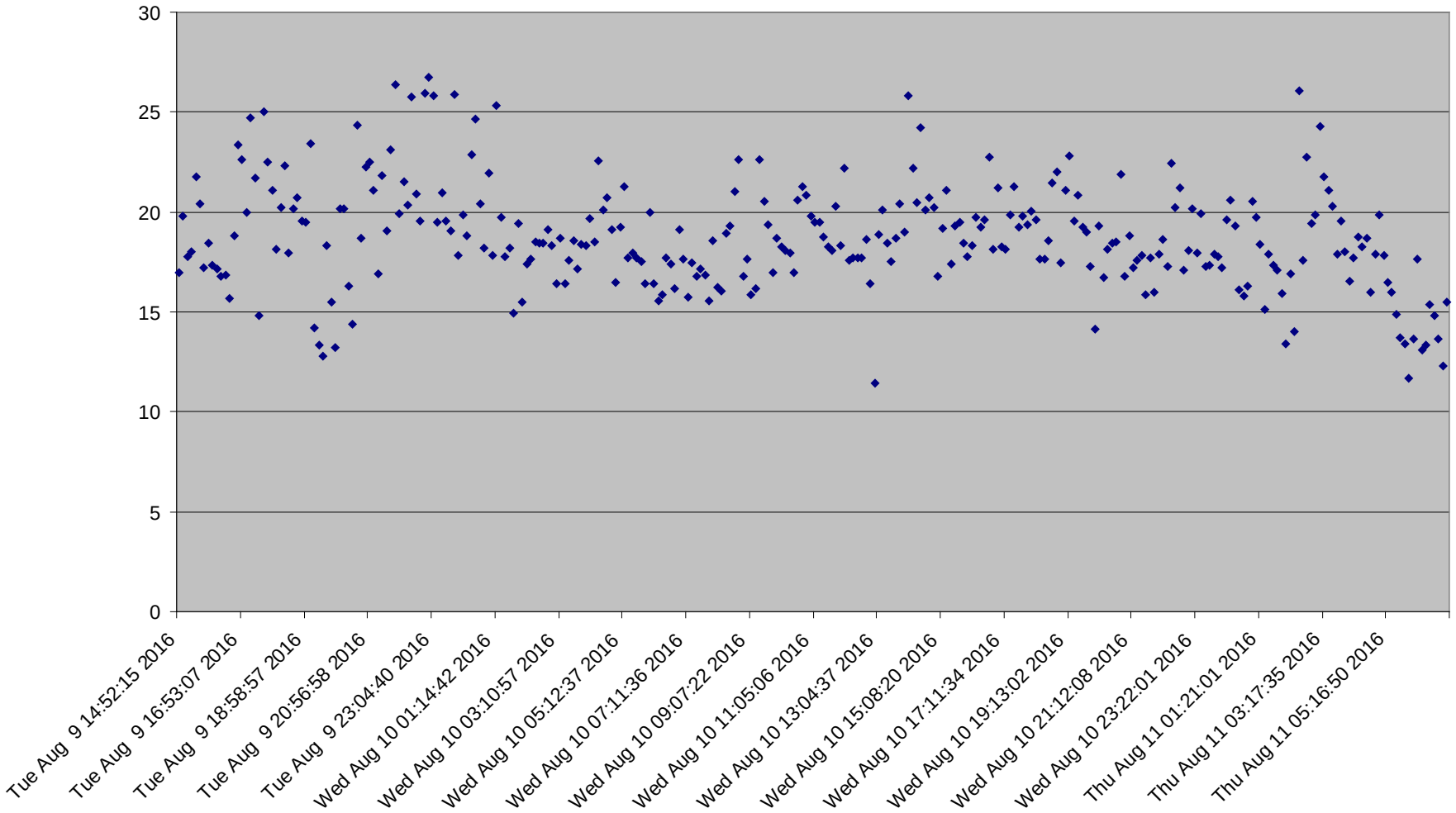
autoflushsize/ splitlevel	10	100	1000
0	0.25±0.00ms (138±3%)	0.21±0.00ms (117±0%)	0.21±0.00ms (117±0%)
1	0.48±0.00ms (268±1%)	0.22±0.00ms (123±1%)	0.21±0.00ms (116±2%)
2	0.51±0.00ms (285±2%)	0.19±0.00ms (106±0%)	0.17±0.00ms (95±2%)
5	0.51±0.00ms (285±3%)	0.19±0.00ms (106±1%)	0.17±0.00ms (95±2%)
99	0.51±0.00ms (284±2%)	0.19±0.00ms (106±1%)	0.17±0.00ms (95±2%)

mDST Write Speed

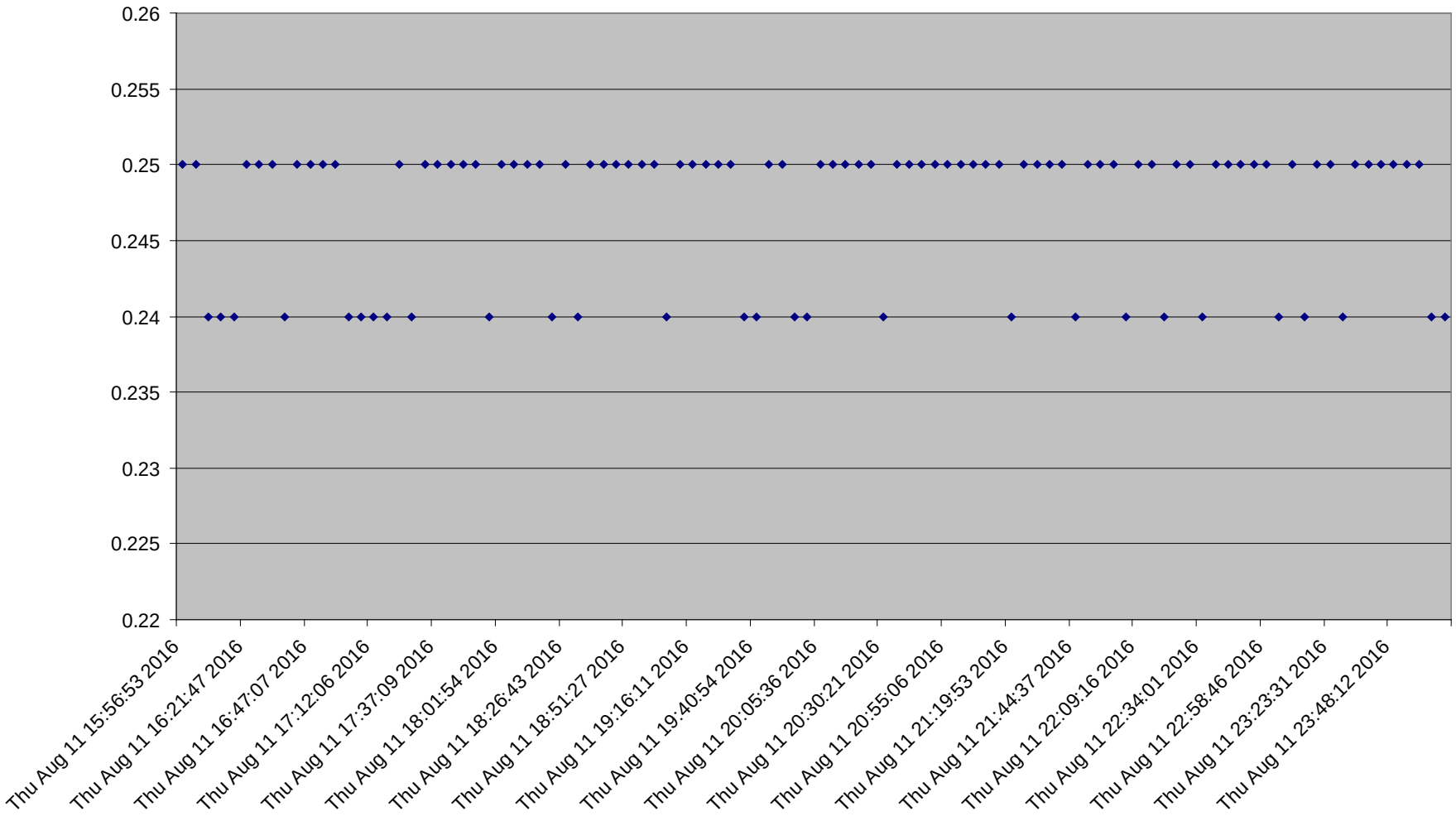
- Speed of writing new mDST normally in basf2
- 100 runs at KEKCC.
- Default (splitLevel=99, autoflushsize=-30000000): 0.47±0.02ms

autoflushsize/ splitlevel	10	100	1000
0	0.76±0.04ms (159±9%)	0.99±0.07ms (210±14%)	0.64±0.04ms (134±7%)
1	1.58±0.05ms (333±10%)	1.10±0.05ms (231±10%)	0.68±0.02ms (143±5%)
2	1.78±0.04ms (375±8%)	1.07±0.03ms (226±7%)	0.65±0.03ms (137±6%)
5	1.78±0.04ms (376±9%)	1.08±0.06ms (227±12%)	0.65±0.03ms (137±6%)
99	1.78±0.05ms (376±11%)	1.08±0.06ms (228±13%)	0.65±0.03ms (138±6%)

Index Read Speed By Local Time



mDST Read Speed By Local Time



mDST as Index

- Test using regular mDST as imput file.
- Works as expected, set “parentLevel” in RootInput module to 1.

parentLevel = 0

Normal mDST read

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	311	0	0.08	0.27 +- 0.20
Gearbox	310	0	0.00	0.00 +- 0.00
ParticleLoader_pi+:loose	310	0	0.00	0.00 +- 0.00
ParticleLoader_K+:loose	310	0	0.00	0.00 +- 0.00
ParticleCombiner_B+ -> K+:loose K-:loose pi+:loose	310	0	2.74	8.84 +- 154.97
MCMatch_B+	310	0	0.00	0.00 +- 0.02
ROEBuilder_B+	310	0	0.00	0.00 +- 0.03
QQBuilder_B+	310	0	0.00	0.01 +- 0.13
NtupleMaker_ntupleFile_mdst_as_index.root	310	0	0.00	0.00 +- 0.00
NtupleMaker_ntupleTree_B+	310	0	0.00	0.01 +- 0.06
Total	311	0	2.84	9.12 +- 154.73

parentLevel = 1

Index read

Name	Calls	VMemory(MB)	Time(s)	Time(ms)/Call
RootInput	311	0	1.06	3.40 +- 15.05
Gearbox	310	0	0.00	0.00 +- 0.00
ParticleLoader_pi+:loose	310	0	0.00	0.00 +- 0.00
ParticleLoader_K+:loose	310	0	0.00	0.00 +- 0.00
ParticleCombiner_B+ -> K+:loose K-:loose pi+:loose	310	0	2.74	8.83 +- 154.74
MCMatch_B+	310	0	0.00	0.00 +- 0.02
ROEBuilder_B+	310	0	0.00	0.00 +- 0.04
QQBuilder_B+	310	0	0.00	0.01 +- 0.13
NtupleMaker_ntupleFile_mdst_as_index.root	310	0	0.00	0.00 +- 0.00
NtupleMaker_ntupleTree_B+	310	0	0.00	0.01 +- 0.06
Total	311	0	3.78	12.16 +- 155.27