

# Report on Status of Metadata system

J.H Kim<sup>1</sup> & S. I Ahn<sup>2</sup> & K. Cho<sup>1</sup>

<sup>1</sup>High Energy Physics Team

<sup>2</sup>e-Science Grid IT Team

KISTI, Daejeon, Korea

Belle II General Meeting, 2010.07.05

## Overview

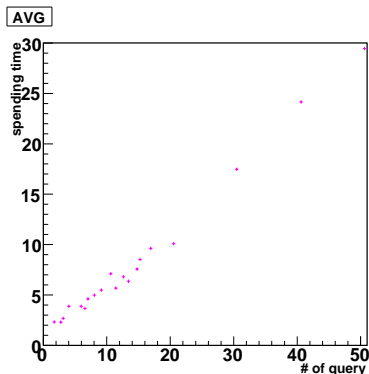
- 1 Before: Large Scale data DH test with Belle data
- 2 Current: scalability test with random generating meta data
- 3 Summary and next plan

## Large Scale data DH test with Belle data

- 1 Size : 31MB, 21 experiments(exp07-exp49), on\_resonance, stream 0,1,2
- 2 Extraction time : 1.8min - 18min/file
- 3 Generating time : 400files/sec
- 4 Performance :
  - UI : hep2.kisti.re.kr
  - Meta system : Melbourne slave( for global network environment )
  - Prototype : belle\_amga\_access
  - Query type : long query( searching all run number)
- 5 Maximum queries : 50

## Performance: with a table (exp07) and multi-queries.

- We perform to search the interesting files with a table of meta-system and changing the number of queries.



- Environment: global network
- Using data: all stream and all run number for exp 07
- The linearity of searching is stable until 50 queries.

# Scalability test with random generating meta data

## ● Generating

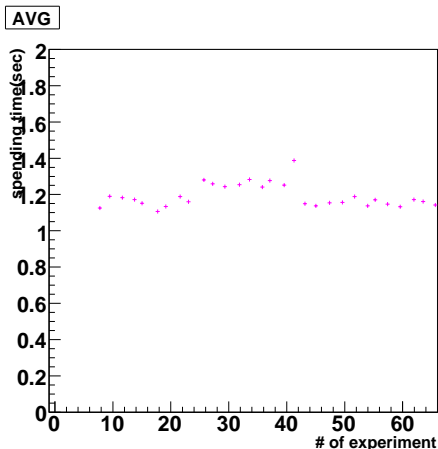
- ▶ Background: Based on TDR
  - Total # of experiments: 30
  - Total # of streams: 6
  - Total # of runs in each experiment: 800
  - Total # of runs in MC types: 4 (uds, charm, charged, mixed)
- ▶ Raw data: 100M files (33.86GBytes)
  - 3.3 M files in each experiment ( = 100 M/30)
  - 4,125 files per run ( = 3.3 M/800)
- ▶ real: 4.3M files files (1.41GBytes)
  - 143K files in each experiment ( = 4.3 M/30 )
  - 180 files in each run ( = 143K/800)
- ▶ MC: 12.5M files files (4.61GBytes)
  - 2.1M files in each stream ( = 12.5M/6 )
  - 70K files in each stream & experiment ( = 2.1M/30 )
  - 17.5K files in each type ( = 70K/4)
  - 88 files in each run ( = 70K/800)

## ● Total size = 39.88 GBytes

## Scalability test with random generating meta data

- 1 We use MC for this test.
- 1 Size : 30 experiments(exp07-exp65), on\_resonance, stream 0,1,2
- 2 Replication: not available
- 2 Generating time : 400files/sec
- 3 Performance Test :  
UI : hep2.kisti.re.kr  
Meta system :slave( 150.183.246.196 )  
Prototype : belle2\_amga\_access  
Query type : Searching all run number for a random experiments
- 2 Maximum queries : 50 and 1000

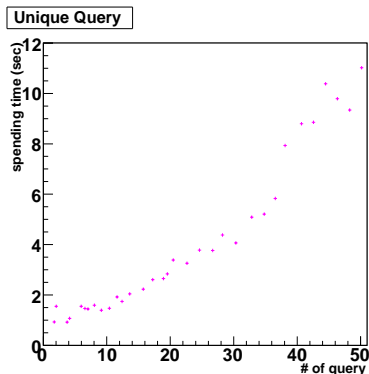
## Performance: searching time for full data of each experiment and stream.



- Environment: local network
- Using data: full data for each exp and stream
- Spending time: 1.1 sec ~ 1.3 sec

## Performance: with a table and multi-queries.

- We perform to search the interesting files with a table of meta-system and changing the number of queries.

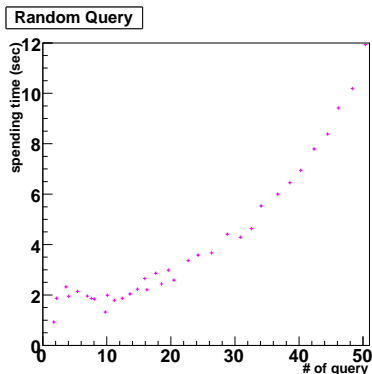


- Environment: local network
- Using data: all run number for each exp and stream
- The linearity of searching is stable until 50 queries.



## Performance: with multi table and multi-queries.

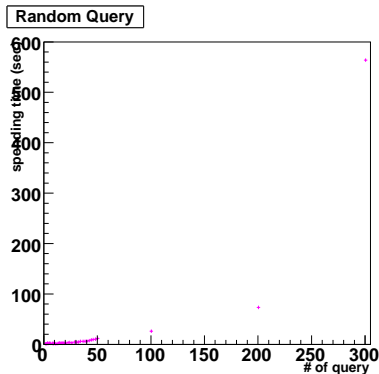
- We perform to search the interesting files with a random table of meta-system and changing the number of queries.



- The linearity of searching is stable until 50 queries and is almost similar to that of using a table and multi-queries.

## Performance: maximum queries are 1000

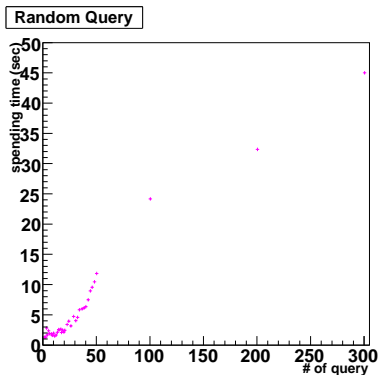
- We perform to search the interesting files with maximum 1000 queries.



- We test the scalability until 300 queries.
- After 300 queries, some queries is not run.

## Performance: maximum queries are 50

- We perform to search the interesting files with maximum 50 queries (DB connection pooling)
- DB connection pooling is a function of AMGA service.



- For huge processes, DB connection pooling give a good result for searching.

## Summary and Next plan

- 1 We are testing the scalability with random generating meta data.
- 2 We are generating the random meta-data for scalability.: done
- 3 multi-query test :  
1<sup>st</sup> : Searching with a table and multi-queries:done  
2<sup>nd</sup> : Searching with random table and multi-queries: done.
- 4 If the meta-system have an overflow, We can reduce the searching time with DB connection pooling.
- 5 We will offer the replication for random meta-data.: will do
- 6 We will test for the global network environment.: will do