



MPII - reading binary files

C. Kleinwort

DESY/HH tracker alignment meeting 26.10.10

Introduction

- ★ Reading binary files from disk

- ▶ Bottleneck for parallelization, ~ 50 MB/s
- ▶ Caching by OS helps, but needs lot of memory

- ★ Gzipped binary files

- ▶ About factor 2 smaller
- ▶ Need time for decompression (zlib)
 - read and decompress in parallel

Parallel reading of binaries

- ★ Loop over binary files

- ▶ Parallelized

- ✦ Each thread reads from different file
 - ✦ After 'end of file' thread takes next (unread) file

- ▶ After first loop

- ✦ Sort files by size (number of records)

- ▶ Second and later loops

- ✦ Read in decreasing order to optimize load balancing
(use smallest files (the loads to be distributed) at the end to have all threads finished at the same time)

Timing example

★ Bows&kinks@UHH

- ▶ X5520, 2.27GHz, 96GB memory, 7 threads
- ▶ 4.6M tracks in 120 files (160 to 160k tracks)
- ▶ Read 19 times (cached by OS) for 6 MP iterations

GZIP	volume* [GB]	#threads (reading)	wall time [h:m]	reading binaries [s]	wait for last thread [s]	reading total time [s]	reading wall time [s]	reading speed [MB/s]
no	353	1	3:49	209	0	209	209	1700
no	353	7	3:47	311	44	355	51	7000
yes	171	1	4:26	2441	0	2441	2441	70
yes	171	7	3:52	2466	134	2601	372	460

* volume by 'du -k', 1GB = 10³MB = 10⁶KB

Summary

- ★ MillePede revision 69

- ▶ Read binary files in parallel: `threads nproc nread`

- ★ Parallelization of reading binaries

- ▶ Decompression is much faster than reading from (usual) disks
- ▶ Size reduction (gzip: factor 2) translates almost into similar time gain

- ★ Tests with disk reading without OS caching

- ▶ Will be done by Frank at PSI