Implementation, evaluation and analysis of Block index for ADIOS

Tzuhsien Wu, Jerry Chou
National Tsing Hua University, Taiwan

Norbert Podhorszki, Yuan Tian
Oak Ridge National Laboratory, USA

Junmin Gu, Kesheng Wu
Lawrence Berkeley National Laboratory, USA

Introduction

- Scientific datasets are commonly stored and managed by parallel file systems and I/O libraries
  - E.g. Lustre, HDF5, NetCDF, ADIOS
  - optimized for reading/writing large chunks of data
  - Data layout and file organization impact query performance
- The characteristics and behaviors of I/O systems should be considered into the design of indexing methods

The idea of "Block index"

- Indexing blocks (consecutive data records) instead of individual data records
    - Reduce index size
    - Reduce number of I/O requests
    - Reading an individual record has similar I/O latency as reading a data block

Implement block index into ADIOS

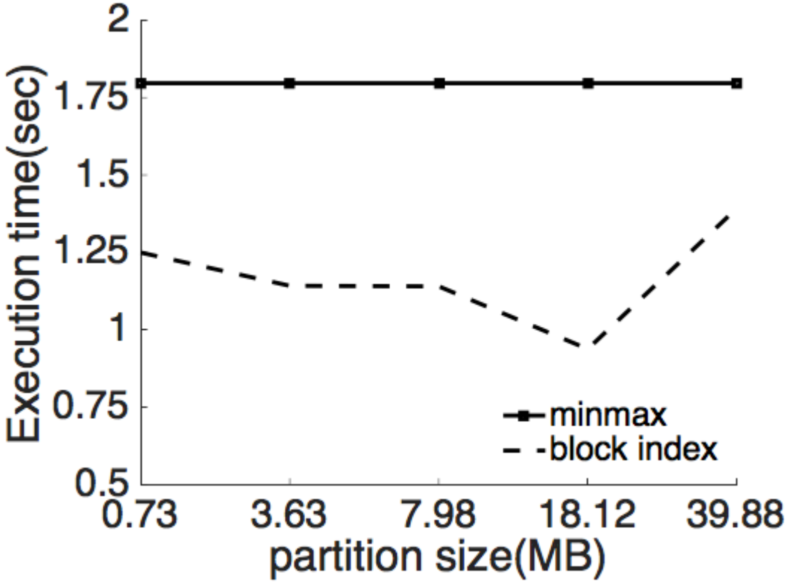
- Minmax method in ADIOS
  - Records the min, max value from each writeblock
  - The size of writeblock => the size of data of each process (can be extremely big)
- Block index method in ADIOS
  - Logically divides a writeblock into smaller partitions
  - Records the min, max values of each partition
  - Using logical partition can maintain the same number of writeblock
  - The I/O requests on the same writeblock can be merged by ADIOS to minimize I/O contention

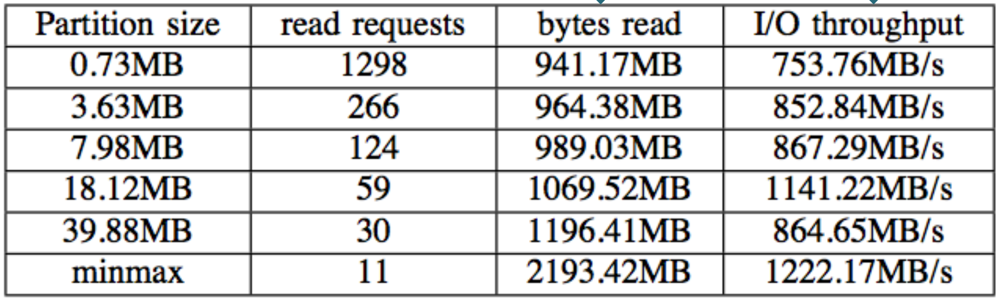
Experiment Setup

- Edison Cray XC30 at NERSC
  - 5576 compute nodes, with 12-core Intel Ivy Bridge 2.4GHz CPU and 64GB memory per node
  - Lustre parallel file system with 72GB peak performance
- S3D dataset
  - Each variable contains 1100*1080*1408 double precision records
  - Each variable is written to file using 64 writeblocks of size 275*270*352 (~200MB)

Performance evaluation

- Varied partition size
  - The performance is a tradeoff between read size and I/O throughput
  - Minmax's read bytes is more than twice the block index

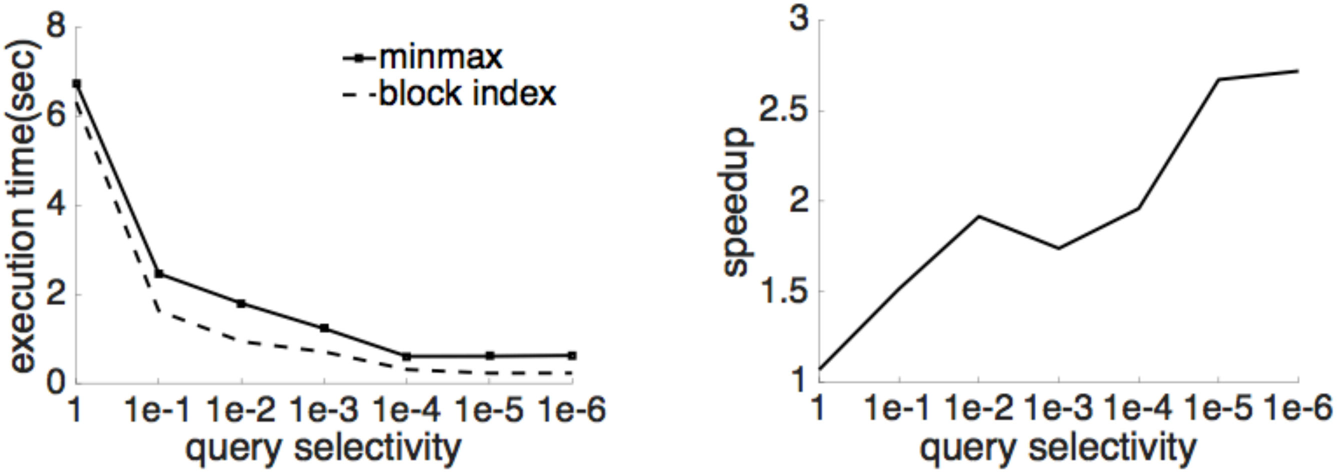




Partition size	read requests	bytes read	I/O throughput
0.73MB	1298	941.17MB	753.76MB/s
3.63MB	266	964.38MB	852.84MB/s
7.98MB	124	989.03MB	867.29MB/s
18.12MB	59	1069.52MB	1141.22MB/s
39.88MB	30	1196.41MB	864.65MB/s
minmax	11	2193.42MB	1222.17MB/s

Performance evaluation

- Varied query selectivity
  - Block index reads less data when query selectivity is smaller => speedup is higher
  - Similar performance under 100% query selectivity



(a) Execution time.

(b) speedup of block index.

Conclusion

- Query performance of minmax is limited by the size of writeblock
- Query performance of Block index that logically partitions a writeblock improves due to less data reading, and more flexible read size
- Future work
  - Performance analysis and modeling of I/O systems
  - Design the algorithm to select the proper block size and request merging condition

THANK YOU