IPDPS 2021 VIRTUAL 17-21 May

iWAPT2021

HLRS

Improving the MPI-IO Performance of Applications with Genetic Algorithm based Auto-tuning

Ayse Bagbaba, Xuan Wang

The Sixteenth International Workshop on Automatic Performance Tuning

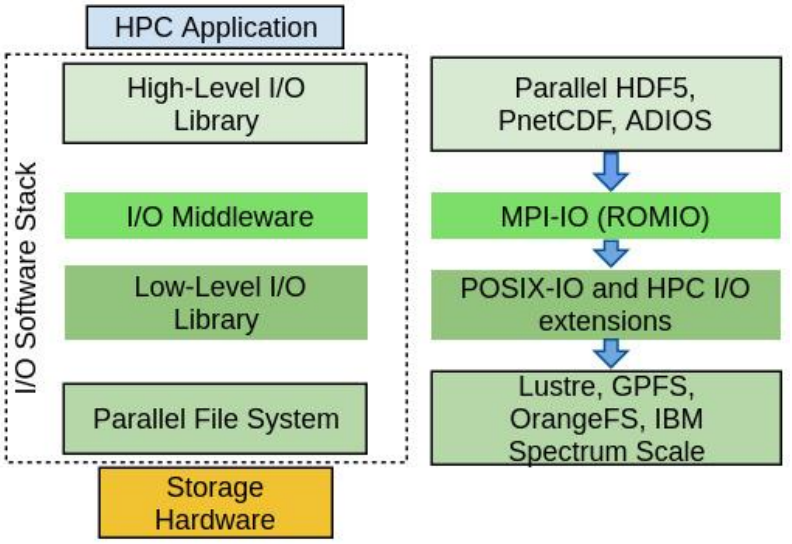
May 21, 2021, Online

Outline

- Motivation & Objectives
- Background
- Genetic Algorithm based Auto-tuning
- Experimental Results
- Conclusion

Motivation & Objectives

- There is no “One Size Fits All” solution to the I/O problem
- Various layers offer **tunable parameters** for improving collective I/O performance
- **Finding good configurations** of an I/O application is often challenging

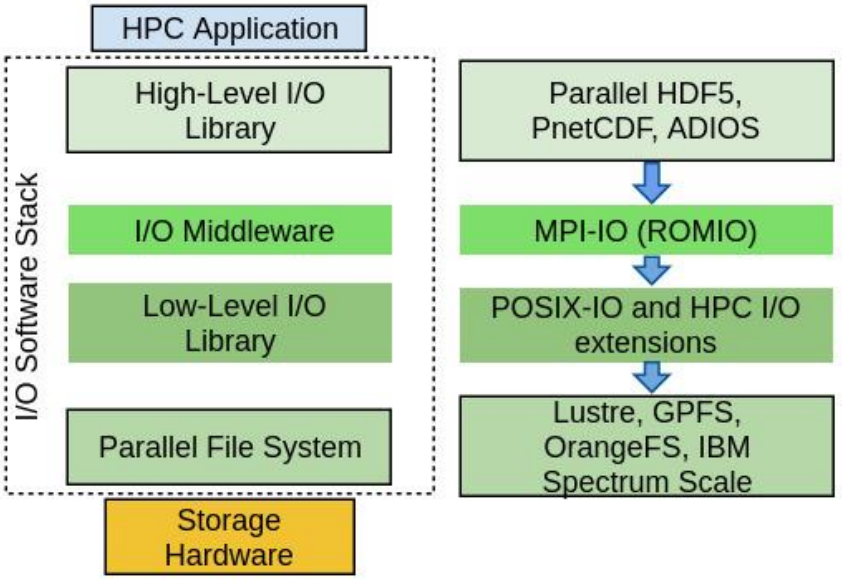


- Manual-tuning, unmanageable task for application developers
- **Solution: Auto-tuning**
- A number of auto-tuning works naïve strategy, heuristic search, machine learning, analytical models etc.

Background

- Performance Factors
  - **romio_cb_write, romio_cb_write, cb_nodes** etc.
  - **stripe_count** (the number of OST)
  - **stripe_size** (the size of one stripe on each OST)

- Barriers
  - How to choose a proper I/O algorithm?
  - Are the I/O algorithms compatible with the underlying parallel file system?



Proposed Approach

Naïve strategy

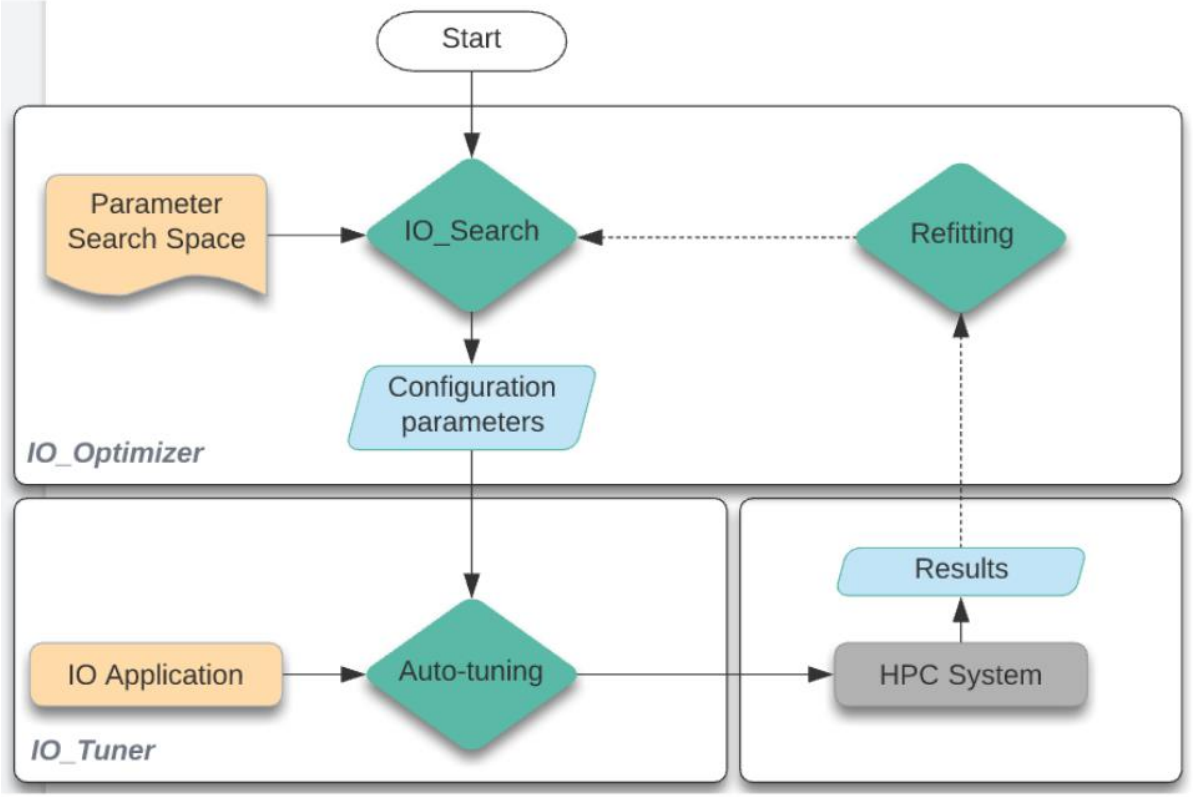
- For selecting tunable parameters
  - execute an application or IO kernel using all possible combinations of tunable parameters for all layers of the I/O stack
- Manual tweaking

Proposed Approaches

- **IO_Search**
  - Adaptive heuristic search-based approach

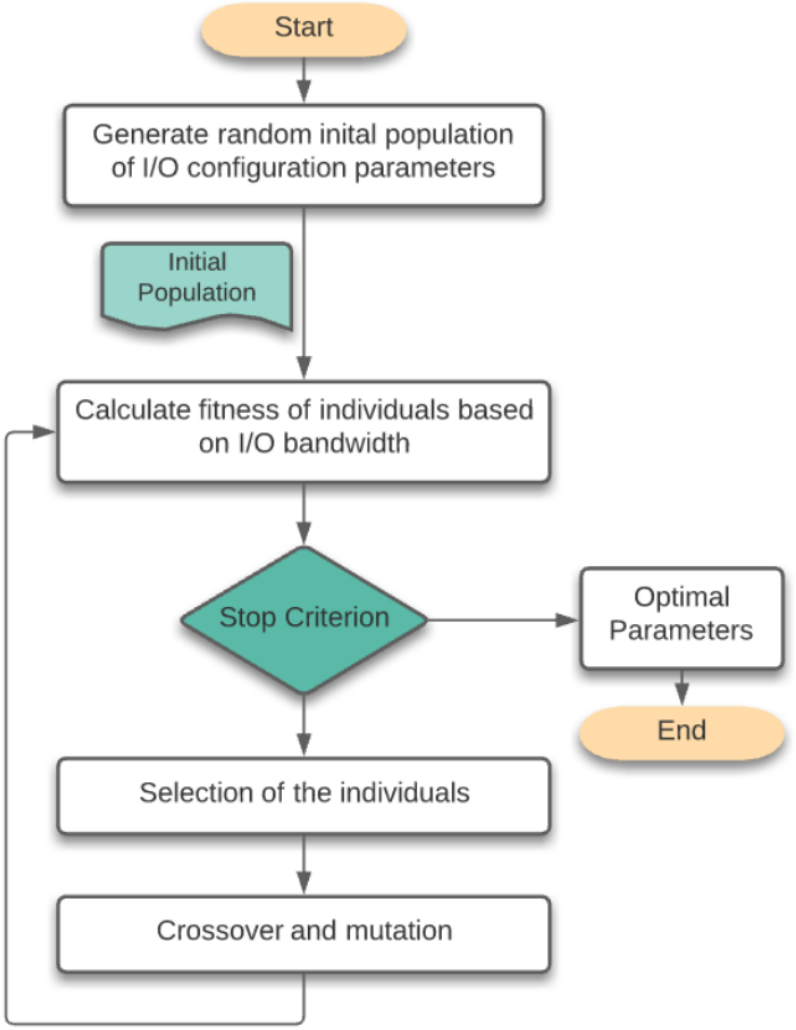
Auto-tuning Parallel I/O

- Search the parameter space with **a small number of tests**
- IO_Search searches the I/O parameter space using a genetic algorithm
- Multiple generations, better parameter combinations!



Genetic Algorithm based Auto-tuning

- Evaluation process:
  - **Start** with randomly selected initial population (group of configuration sets)
  - **Pass through** successive generations
  - **Use** the **I/O bandwidth** as the **fitness** evaluation for a given I/O configuration
  - **Evaluate** the fitness of the population
  - **Consider** the fastest I/O configurations for inclusion in the next generation
  - A series of **mutations**
  - **Store and tune best-performing I/O** configuration



Implementation

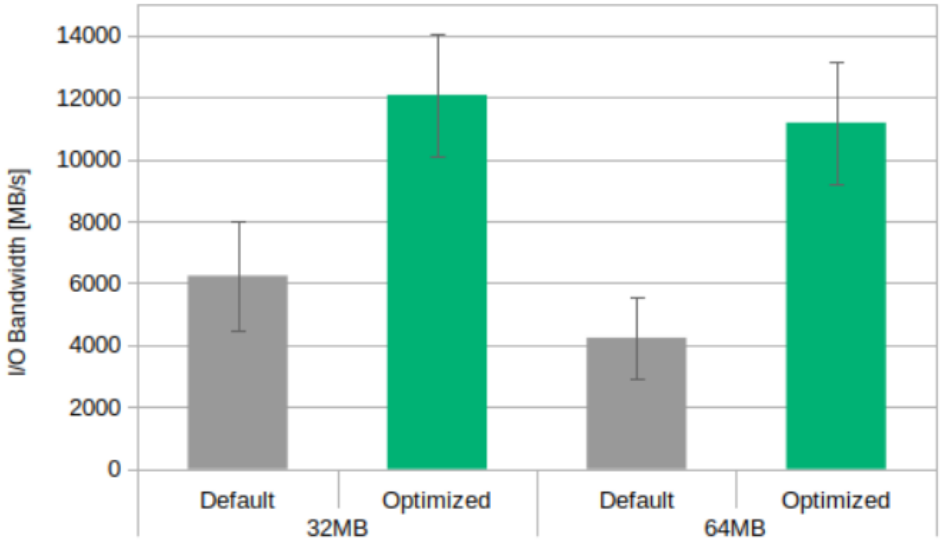
- Important configuration parameters for Lustre and MPI-IO layer-ROMIO library

Name	Value
number of cores	64 - 1200
number of bytes	256 KB - 64 MB
number of aggregators	1 - 16
striping count	1 - 16
striping unit	1 MB - 16MB
collective I/O	automatic; disable; enable
I/O pattern	collective, individual

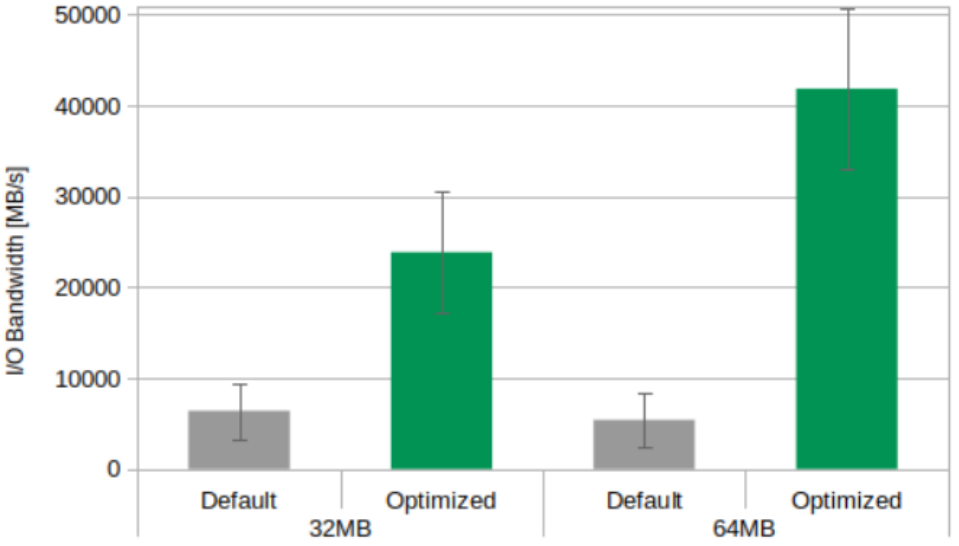
- mutation rate: 15%
- population size: 10
- the fitness value: the I/O bandwidth
- number of generations: 30

- Benchmarks:
  - **IOR**
    - configured in the range from 32 MB to 64 MB block sizes to write in the shared files collectively
  - **MPI-Tile-IO**
    - configured number of tiles as much as number of cores, and in the range from 8096KB to 16184 KB element sizes

Results, IOR

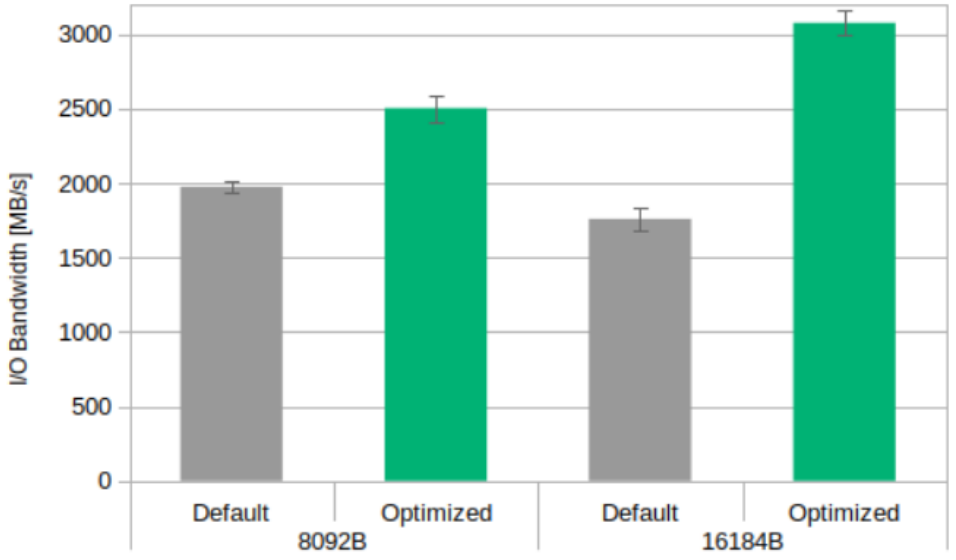


(a) 240 cores

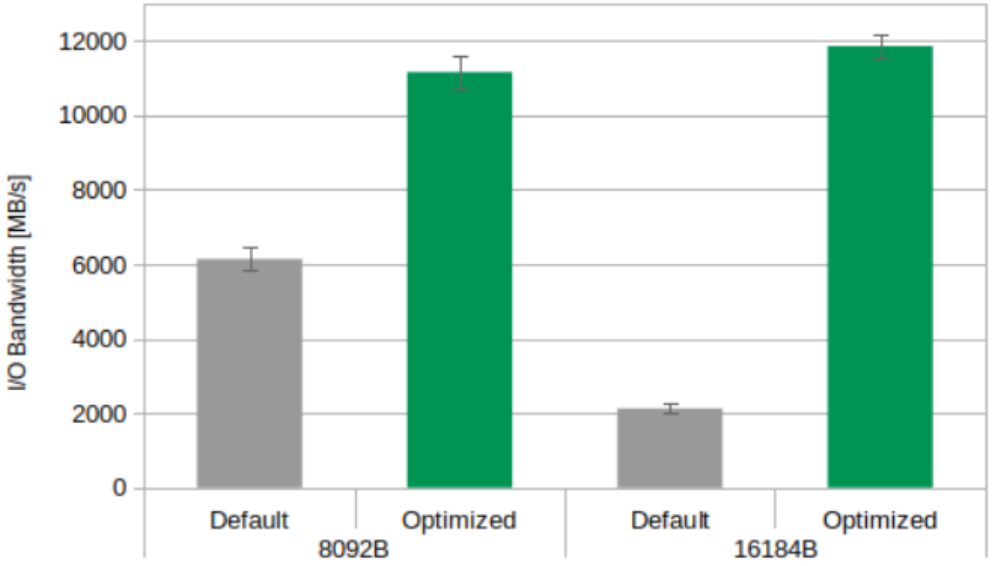


(b) 1200 cores

Results, MPI-Tile-IO



(a) 64 cores



(b) 256 cores

Results

- **Naïve strategy**
  - Evaluation of **all** configurations to find the best
  - ~40 hours

- **GA based auto-tuning approach**
  - ~2.5 hours and ~4 hours for the two concurrencies in IOR,
  - ~1.0 hours and~1.5 hours for the two concurrencies in MPI-Tile-IO

- A more comprehensive training data set can give better prediction results

Application		IOR (MB/s)		MPI-Tile-IO (MB/s)	
#Cores		240	1200	64	256
Use case 1	Default	6238.56	6402.08	1974.462	6138.917
	Tuned	12075.01	23859.15	2503.22	11257.42
	Speedup	1.93	3.73	1.27	1.83
Use case 2	Default	4238.57	5412.91	1759.326	2122.027
	Tuned	11186.23	41859.44	3075.17	11859.23
	Speedup	2.64	7.74	1.75	5.59

Conclusion

- GA based I/O auto-tuning approach to tune the parallel I/O stack parameters
- Improvements for write bandwidths in main HPC I/O benchmarks on supercomputer Vulcan
- The overhead to find the best parameters is drastically reduced
- For any benchmark or I/O application, our model can be applied with negligible effort
- Can be understood by users with little knowledge of parallel I/O without any post-processing step
- The parameters discussed are system dependent, but new parameters can be easily integrated to the configuration files

IPDPS 2021 VIRTUAL 17-21 May

iWAPT2021

HLRS

Thanks

bagbaba@hlrs.de

xuan.wang.51@ gmail.com