Synthetic Parallel Applications

Rosie Wacha, Darrell Long
Department of Computer Science
UC Santa Cruz

Gary Grider, James Nunez
HPC-5: HPC Systems Integration
Los Alamos National Laboratory

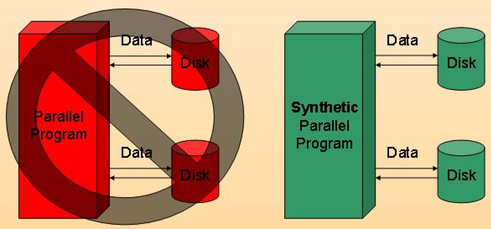
LA-UR-06-6337

Storage system design and optimization require usage data in order to determine where in the system to focus resources. Typically this usage data is created by observing the response of the system while running synthetic benchmarks designed to place high stress on particular aspects of the system. The problem with many benchmarks is that they encourage improvements to areas of the system that do not improve actual performance under typical load or they do not capture the true workload of important applications. Ideally, the benchmarks should be similar to actual user applications so that the storage system can be designed and optimized for those applications.

User applications often contain data that is private due to business or national security reasons so the applications cannot be released to researchers or storage vendors. The I/O Extractor runs a possibly classified user application while creating logs of the I/O events that occurred during its execution. Our Synthetic Parallel Application Generator (SPAG) analyzes this log, creates a model of the I/O behavior of the original program, and writes a new application that mimics that I/O. The SPAG ignores scientific computation and program structures and the Synthetic Parallel Application recreates I/O events with random data. Any application can be run through the tool to generate a synthetic benchmark, thus allowing release of and testing against the real I/O workload to vendors and universities.

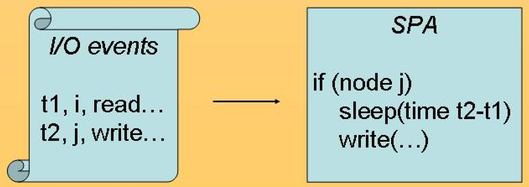
INTRODUCTION & MOTIVATION

- Researchers need I/O data from real user applications to verify new designs, but it is often difficult to obtain
- LANL's scientific applications are classified
- SPA is created based on all I/O the application performs and is not classified



SYNTHETIC PARALLEL APPLICATION

- Mimic the user I/O of a single run of a real parallel application
- Parse system calls performed by original program into events
- Sort events by time and treat as independent
- Parallel program is a sequential list of events with conditionals specifying the node that should perform that event
- Ignore nonuser I/O (such as loading libraries)

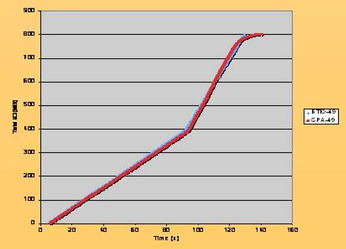


RESULTS & DISCUSSION

- To verify the validity of the SPA generator, we created SPAs for two benchmark programs
- Applications were run on 164-node Linux cluster
- For large time scales, the I/O of the SPA accurately matches the I/O of the original parallel application

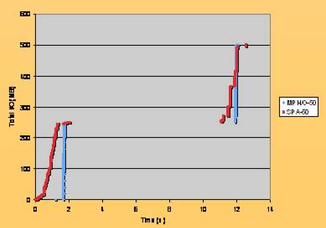
BTIO from NAS Parallel Benchmarks 3.2.1

- Computation with periodic write dumps followed by verification (read) phase
- 49 nodes
- 800 MB total I/O (reads and writes)
- Total run time 140.5 s



MPI I/O Test benchmark from LANL

- Write phase, then 10 second sleep, and read phase
- 50 nodes
- 500 MB total I/O (reads and writes)
- Total run time 12.6 s
- SPA matches within 1-2 s
- Initial mismatch of SPA is due to ignoring non-user I/O



CONCLUSION

We can create SPAs that are accurate for large time scales of long-running benchmarks

FUTURE WORK

Short term goals
- Improve accuracy for short running programs so we can better model bursty I/O
- Measure and correct for clock skew across nodes
- Try on real scientific applications
- Port to other clusters and compare results of I/O event durations on different storage configurations

Long term goals
- Statistical analysis of runs of original application to create SPA
- Create SPAs to analyze viability of future storage setups for user applications

ACKNOWLEDGMENTS

- Research funded in part by a U.S. Department of Education GAANN fellowship
- I thank my advisors and the UCSC Storage Systems Research Center for valuable feedback and advice