T2M: Converting I/O Traces to Workload Models

Vasily Tarasov (student, presenter), Koundinya Santhosh Kumar (student)
and Erez Zadok (Stony Brook University)
Geoff Kuenning (Harvey Mudd College)

I/O traces contain valuable information about the workloads encountered in production environments. By studying a trace one can optimize a system for the specifics of the workload and as a result attain higher performance levels. Traces are also useful for the proper performance evaluation of the existing systems. Out of 68 storage and file system papers published in 2010 in major system conferences (FAST, OSDI, HotStorage, MSST, and ATC), 35 used standard or custom traces for their performance evaluation. It is therefore not surprising that SNIA's trace repository grows rapidly and contains at the moment more than 20 traces collected by different organizations [5].

Although traces are indispensable for proper performance evaluation in some projects, they are cumbersome to replay and do not scale well as systems become more powerful. Conversely, synthetic benchmarks are easy to use and can scale the load efficiently. A lot of modern benchmarks have rich and extensive languages to describe workloads: Filebench [6], FFSB [4], fio [2], SynRgen [1] and others. The languages of these benchmarks can express not only micro-workloads (useful for analysis of the specific system properties), but can describe real-world workloads as well. For example, Filebench contains Fileserver, Web-server, OLTP, and several other workloads that were used for performance evaluation in 3 papers in the recent FAST 2010 conference.

But how do we create a model that closely matches the workload observed in the real life? Nowadays it is usually done manually using some intuitive assumptions about the applications' or users' behavior. Traces contain all the information we need to generate an accurate model of the workload. Before, researchers analyzed traces *manually* to create workload models. The goal of the T2M project is to design a methodology and a set of tools that can automate the process of converting traces to concise and accurate workload models.

There are three main directions in this project. First, we are interested in determining a complete set of parameters that need to be extracted from a trace; e.g., the I/O size is intuitively an important parameter, so its statistical distribution should be extracted. But should we also extract data entropy of the I/O buffers from the trace? For the systems that use compression it is an important parameter; for other systems it is not. There are also certain dependencies between parameters that need to be taken into account. For example it makes sense to distinguish I/O size distributions for read requests from I/O size distributions for write requests. The more parameters are considered, the more inter-dependencies can be observed between them and the more complex workload models will be generated.

The second direction of this research is determining at which level (system call, file system, block, etc.) the traces need to be collected to be able to generate an accurate model. Can we benefit from using the traces from different levels to increase the accuracy of the corresponding model? At the moment we collect the traces and generate the workloads at the file system level, but we are building a system that is capable of simultaneously collecting traces at the system call, block, and NFS levels.

Finally, at the moment it is not clear if the languages provided by the existing benchmarks are rich enough to express the complexity enclosed in the traces. Currently, our target language is Filebench's Workload Model Language (WML). So, we are extending Filebench's language gradually. One of the features we have recently added allows users to dynamically load shared objects/modules with implementations of custom statistical distributions. We ported Mersenne Twist Library [3] (which supports about two dozens of different distributions) to the developed API. This feature allows us to generate parameter's values with more complex distributions observed in real-world traces.

Summary. In this project we investigate the possibility to automatically create workload models from existing traces. This will provide the scalability and the ease of use of the synthetic benchmarks while maintaining the realism of the traces.

References

[1] M. Ebling and M. Satyanarayanan. SynRGen: An Extensible File Reference Generator. In *Proceedings of ACM SIGMETRICS Conference*, 1994.

[2] fio - Flexible I/O tester, 2010. `http://freshmeat.net/projects/fio/`.

[3] G. Kuenning. Mersenne Twist Pseudorandom Number Generator Package, 2010. `http://lasr.cs.ucla.edu/geoff/mtwist.html`.

[4] J. Santos and S. Rao. FFSB: Flexible file system benchmark. `http://sourceforge.net/projects/ffsb/`.

[5] SNIA. SNIA - storage network industry association: IOTTA repository, 2007. `http://iotta.snia.org`.

[6] Sun Microsystems. Filebench. `www.solarisinternals.com/si/tools/filebench`.