Highlights of X-Stack ExM[1] Deliverable: MosaStore

Matei Ripeanu— July 20, 2016
matei@ece.ubc.ca

MosaStore is an exploratory part of the X-Stack project[1] *"ExM: System support for extreme-scale, many-task applications"*. The ExM project proposed to use concurrent workflows supported by the Swift language and runtime as an innovative programming model to exploit parallelism in exascale computers. MosaStore aims to support this endeavour by improving storage support for workflow-based applications, more precisely by exploring the gains that can be obtained from co-designing the storage system and the workflow runtime engine. MosaStore has been developed primarily at the University of British Columbia.

Key Idea Explored

Our thesis is that custom metadata can be used as a bidirectional communication channel between workflow applications and the storage system and thus become the key enabler for co-designing the storage and the workflow runtime engine thus unlocking cross-layer optimizations that, today, are hindered by an ossified file-system interface. This communication channel is *bidirectional* as the cross-layer optimizations enabled are based on information passed in both directions across the storage system interface (i.e., application to storage and storage to application). Possible cross-layer optimizations include:

- (*top-down*) Applications can use metadata to provide hints to the storage system about their future behavior, such as: per-file access patterns, ideal data placement (e.g., co-usage), predicted file lifetime (i.e., temporary files vs. persistent results), access locality in a distributed setting, desired file replication level, or desired quality of service. These hints can be used to optimize the storage layer.
- (*bottom-up*) The storage system can use metadata as a mechanism to expose key attributes of the data items stored. For example, a distributed storage system can provide information about data location, thus enabling location-aware scheduling.

The proposed mechanism has four interrelated advantages: it uses an application-agnostic mechanism, it is incremental, it offers a low cost for experimentation, and it focuses the research community effort on a single storage system prototype, saving considerable development and maintenance effort dedicated, nowadays, to multiple storage systems each targeting a specific workload. First, the communication mechanism we propose: simply annotating files with arbitrary <*key, value*> pairs, is application-agnostic. Second, our approach enables evolving applications and storage-systems independently while maintaining the current interface (e.g., POSIX), and offers an incremental transition path for legacy applications and storage-systems: A legacy application will still work without changes (yet will not see performance gains) when deployed over a new storage system that supports cross-layer optimizations. Similarly a legacy storage will still support applications that attempt to convey optimization hints, yet it will not offer performance benefits. As storage and applications incrementally add support for passing and reacting to optimization hints, the overall system will see increasing gains. Finally, exposing information between different system layers implies tradeoffs between performance and transparency. To date, these tradeoffs have been scarcely explored. A flexible encoding (key/value pairs) as the information passing mechanism offers the flexibility to enable low-cost experimentation within this tradeoff space.

Approach

We demonstrated our approach by designing and prototyping MosaStore, a POSIX-compatible storage system that is instantiated on-the-fly to aggregate the resources of the computing nodes allocated to a batch application (*e.g.*, disks, SSDs, and memory) and offers a shared file-system abstraction optimized for workflow applications. This is based on two key features. First, it optimizes the data layout (e.g., file and block placement, file co-placement) to efficiently support the workflow data access patterns (as hinted by the application). Second, the storage system uses custom metadata to expose data location information so that the workflow runtime engine can make location-aware scheduling decisions. These two features are key to efficiently support workflow applications as their generated data access

[1] PIs for the project *"ExM: System support for extreme-scale, many-task applications"* are: Michael Wilde (Argonne National Laboratory), Daniel S. Katz (University of Chicago), Matei Ripeanu (University of British Columbia), Rusty Lusk (Argonne National Laboratory), and Ian Foster (University of Chicago).

patterns are irregular and application-dependent.

NOTABLE RESULTS.

Our experience demonstrates that it is feasible to have a POSIX compatible storage system that can be yet optimized for each workflow application (or application mix) even if the workflows have a different access pattern for different files. The key contributions of this work are:

- *We propose a new approach* that uses custom metadata to enable cross-layer optimizations between applications and the storage system. Further, we argue that this approach can be adopted incrementally. This suggests an evolution path for co-designing POSIX-compatible file-systems together with the middleware ecosystem they coexist such that performance efficiencies are not lost and flexibility is preserved, a key concern to support legacy applications. [1, 3, 11]
- *We have developed MosaStore – a prototype system with an extensible storage system architecture* that supports cross-layer optimizations. We have demonstrated, using synthetic benchmarks as well as real-world workflows, that this design can bring sizeable performance gains. [1, 3, 6]
- *We have integrated MosaStore with Swift* (and other workflow schedulers) and demonstrated that MosaStore achieves its goals, with only minor changes to the workflow scheduler, and without changing the application code or tasking the developer to annotate their code to reveal the data usage patterns. [3, 6, 11]
- *We have proposed ways to automate the finding of an optimal configuration of the storage system* (in terms of performance or energy) [2, 4, 5, 7]

FURTHER INFORMATION

All MosaStore documentation, code, and related publications can be found at the following locations:

- MosaStore home page: http://www.mosastore.net
- NetSysLab homepage: http://netsyslab.ece.ubc.ca
- Matei Ripeanu's homepage: http://www.ece.ubc.ca/~matei/

PEER-REVIEWED PUBLICATIONS DETAILING MOSASTORE EXPERIENCE

[1] *A Software-Defined Storage for Workflow Applications*, Samer Al-Kiswany, Matei Ripeanu, IEEE Cluster Conference (Cluster'16), Taipei, Taiwan, September 2016 pdf

[2] *Support for Provisioning and Configuration Decisions for Data Intensive Workflows*, Lauro Beltrão Costa, Hao Yang, Samer Al-Kiswany, Matei Ripeanu, IEEE Transactions on Parallel and Distributed Systems (TPDS), accepted July 2015 pdf

[3] *The Case for Workflow-Aware Storage: An Opportunity Study*, Lauro Beltrao Costa, Hao Yang, Emalayan Vairavanathan, Abmar Barros, Kethan Maheshwari, Gilles Fedak, Daniel Katz, Michael Wilde, Matei Ripeanu, Samer Al-Kiswany, Journal of Grid Computing, Springer, accepted June 2014. pdf

[4] *Energy Prediction for I/O Intensive Workflow Applications*, Hao Yang, Lauro Beltrao Costa, Matei Ripeanu, 7th ACM Workshop on Many-Task Computing on Grids and Supercomputers, collocated with SC'14, New Orleans, LA, November 2014 pdf

[5] *Supporting Storage Configuration for I/O Intensive Workflows*, Lauro Beltrão Costa, Samer Al-Kiswany, Hao Yang, Matei Ripeanu, ICS'14 2014 ACM International Conference on Supercomputing, June 2014, Muenchen, Germany paper

[6] *Evaluating Storage Systems for Scientific Data in the Cloud*, Ketan Maheshwari, Justin M. Wozniak, Hao Yang, Daniel S. Katz, Matei Ripeanu, Victor Zavala, Michael Wilde, 5th Workshop on Scientific Cloud Computing (ScienceCloud), collocated with HPDC 2014, Vancouver, Canada June 2014. **(*Best Paper Award*)** pdf

[7] *Predicting Intermediate Storage Performance for Workflow Applications*, Lauro Beltrão Costa, Samer Al-Kiswany, Abmar Barros, Hao Yang, Matei Ripeanu, 8th Parallel Data Storage Workshop (PDSW) collocated with SC'13, November 2013, Denver. paper

[8] *A Workflow-Aware Storage System: An Opportunity Study*, Emalayan Vairavanathan, Samer Al-Kiswany, Lauro Costa, Zhao Zhang, Daniel Katz, Michael Wilde, Matei Ripeanu, 12th International Symposium on Clusters, Cloud, and Grid Computing (CCGrid'12), Ottawa, Canada, May 2012 (acceptance rate: 27%) pdf

[9] *ExM: High Level Dataflow Programming for Extreme-Scale Systems*, Timothy G. Armstrong, Justin M. Wozniak, Michael Wilde, Ketan Maheshwari, Daniel S. Katz, Matei Ripeanu, Ewing L. Lusk, Ian T. Foster, 4th USENIX Workshop on Hot Topics in Parallelism (HotPar '12), June 7-8, Berkeley, CA.

[10] *AME: An Anyscale Many-Task Computing Engine*, Zhao Zhang, Daniel Katz, Matei Ripeanu, Mike Wilde, Ian Foster, 6th Workshop on Workflows in Support of Large-Scale Science (collocated with SC 2011), Seattle, November 2011 pdf

[11] *A Cross-Layer Optimized Storage System for Workflow Applications*, Samer Al-Kiswany, Lauro B. Costa, Hao Yang, Emalayan Vairavanathan, Matei Ripeanu, Special Issue on Workflows for Data-Driven Research, Journal Future Generation Computer Systems, submitted April 2016.