Scalable Data Middleware for Smart Grids

Jian Yin, Ian Gorton
Pacific Northwest National Laboratory
{jian.yin, ian gorton}@pnl.gov

Smart grids promise to improve the efficiency of power grid systems and reduce green house emissions through incorporating power generation from renewable sources and shaping demands to match the supply. Renewable sources include solar or wind. Power generation from these sources is affected by weather factors that can be highly fluctuating. To ensure these energy sources can be utilized efficiently, smart grid systems often shape demand through incentive to match the supply. As a result, the whole system becomes highly dynamic and requires constant adjusting. How to adjust the system can have a great impact on the efficiency and reliability of power grid systems, which offer many opportunities for innovation. In the previous work by us and other researchers [2, 4, 5, 7, 10, 11], we have identified and developed several applications can be used to optimize power grid operations.

However, these applications rely on the precise estimation of the state of power grid systems. To enable precise estimate of power grid, enormous amount of data from millions of sensors from power grid systems must be used. Moreover, the relevant data must be delivered to applications within real time constraints. Even though millions of sensors such as phase measurement units (PMU) and smart meters are being widely deployed over the Internet, these does not exist a software system can collect, store, retrieve, and deliver these amount of data in real time.

OSIsoft PI [6], which is built on top of SQL server, is the most often used system in power companies. Even though PI provides sufficient data retrieval capability for current power grids, which involves a limited number of supervisory control and data acquisition(SCADA) sensors generating data in low temporal resolution, it cannot handle the orders of magnitude increase in the number of sensors and data volume for the emerging smart grids.

Even though there are quite a few scalable systems [1, 3, 9] built for Internet services, these systems are built for different applications whose design requirements and workload characteristics are much different from smart grids systems. Most of those systems must handle arbitrary insertion, deletion and modifications; some of those systems must handle a wide of ranges of data retrievals; and yet another set of these systems are specially customized to run on tens of thousands of commodity machines that partial failures are frequent. Those systems are often built on top of high-level system interfaces such as file system interfaces or database interfaces. As a result, much indirection is introduced into these systems, which can cause both high overhead and unpredictability due to alternative execution paths.

The special requirements for smart grids include near real time response time, which require the latency for data ingestion, retrieval, and delivery operations to be as predicable as possible. Moreover, the cost can be a major factor that can affect deployment. Thus, to utilize the hardware as efficiently as possible is essential.

We designed GridMW, a scalable near real time data middleware to meet these design requirements. By tailoring the design of GridMW to the specific characteristics of power grid data and applications, we are able to reduce overhead and unpredictability associated with indirection in high level system interfaces without introducing much complexity in implementing with low level system interfaces. Power grid data are characterized with frequent insertion, less frequent deletion, and rare updates. Moreover, the power grid data are often highly structured and also are inserted into our system with temporal locality. Additionally, there is only a limit set of data retrieval operations. We can focus on specialized data storage and load balancing techniques to speed up the limit set of operations.

We design a specialized log structure inspired storage structure to store our data. The data are written directly to block device through block device interface. This eliminates much indirection, high overhead, and unpredictability associated with high level system interface such file system interfaces, in which one access can lead to multiple disk accesses for reading inode blocks, reading a variable number of indirect block, and reading the data block itself. We designed specialized data structures to keep track of data and free space in the storage devices. By leverage that the data are often inserted in temporal order, we are able to minimize the metadata to the point that can be easily keep in the main memory. Hence, in our system, a data insertion is translated into only one disk access. Our system allows customized index for data retrievals. We aggregate local main memory, SSD and remote main memory and SSD to enable most metadata lookups to happen in the types of memory that support random accesses. Remote direct memory access (RDMA) allows us to utilized remote RAM and SSD efficiently. We are able to reduce data retrieval overhead to one disk access in most of the cases, which allows us to meet latency requirements in the order of milliseconds. Preliminary results show that for some workloads, we can improve performance by an order of magnitude compared to traditional systems.

References

1. B. F. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H. -A. Jacobsen, N. Puz, D. Weaver, and R. Yerneni. PNUTS: Yahoo!'s hosted data serving platform. In Proceedings of the 34th International Conference on Very Large Data Bases (VLDB'08), pages 1277-1288, August 2008.
2. J. Nieplocha, D. Chavarra-Miranda, V. Tipparaju, Z. Huang, and A. Marquez. 2008. "A parallel WLS state estimator on shared memory computers." In Proceedings of The 8th International Power Engineering Conference, IPEC 2007, Piscataway, NJ, pp. 395-400.
3. G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. 2007. Dynamo: amazon's highly available key-value store. In Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles (SOSP '07). ACM, New York, NY, USA, pp. 205-220.
4. Z. Huang, and J. Nieplocha. "Transforming Power Grid Operations via High Performance Computing." In 2008 IEEE Power Engineering Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century. IEEE , Piscataway, NJ.
5. I. Gorton, Z. Huang, Y. Chen, B. Kalahar, S. Jin, D. Chavarría-Miranda, D. Baxter, and J. Feo, "A High-Performance Hybrid Computing Approach to Massive Contingency Analysis in the Power Grid", the 5th IEEE International Conference on e-Science, Oxford, UK, Dec 7-9th, 2009, pp. 277–283.
6. OSISoft PI System. http://www.osisoft.com/.
7. Z. Huang, Y. Chen, and J. Nieplocha, "Massive Contingency Analysis with High Performance Computing", Proceeding of IEEE PES General Meeting, Calgary, Canada, July 26-30, 2009, pp. 1-8.
8. P. Cudre-Mauroux, H. Kimura, K.-T. Lim, J. Rogers, R. Simakov, E. Soroush, P. Velikhov, D. L. Wang, M. Balazinska, J. Becla, D. DeWitt, B. Heath, D. Maier, S. Madden, J. Patel, M. Stonebraker, and S. Zdonik. 2009. A demonstration of SciDB: a science-oriented DBMS. Proc. VLDB Endow. 2, 2 (August 2009), 1534-1537.
9. F. Chang, J. Dean, S. Ghemawhat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, and R.Gruber. Bigtable: A distributed storage system for structured data. In OSDI'06, pages 205–218, November 2006.
10. Y. Chen, Z. Huang, P. Wong, P. Mackey, C. Allwardt, J. Ma, and F. Greitzer. 2010. "An Advanced Decision Support Tool for Electricity Infrastructure Operations." In Critical Infrastructure Protection IV, IFIP Advances in Information and Communication Technology, vol. 342, no. 2010, ed. T. Moore and S. Shenoi, pp. 245-260.
11. Z. Huang, N. Zhou, F. Tuffner, Y. Chen, D. Trudnowski, W. Mittelstadt, J. Hauer, and J. Dagle. 2010. "Improving Small Signal Stability through Operating Point Adjustment." In Proceedings of the 2010 IEEE Power and Energy Society General Meeting.