Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

STREAMING DATA MANAGEMENT FOR THE ONLINE PROCESSING OF SIMULATION DATA

Johannes Schützel
Jan Himmelspach
Holger Meyer
Andreas Heuer
Adelinde M. Uhrmacher

Albert Einstein Str. 22
University of Rostock
18059 Rostock, GERMANY

ABSTRACT

The fast processing (storing; computing control variates) of simulation data is essential for efficient simulation. The impact of poor data processing might annihilate all the efforts invested into designing high-performance algorithms for computing the trajectories. Streaming data management can be used as an alternative to the more classical approaches (in-memory and write-and-read-back mechanisms). In first experiments our implementation of streaming data management shows better run time performance than in-memory solutions, although the streaming additionally integrates online calculations, e.g., to determine variance and average.

1 STREAMING DATA MANAGEMENT FOR SIMULATION DATA

In Himmelspach et al. (2010) we have shown that already the storing of simulation data has a significant impact on the overall runtime of a simulation. These experiments did not take into account a second important aspect about data handling: the simulation data computed might be needed for instant simulation control (i.e., replication end estimation; number of replication; etc.). Thus, to only consider the writing operation is not sufficient for many simulation studies. Depending on the solutions in existing M&S software products data might be kept in memory, might be discarded or stored, or might be written to a data sink and read back for the processing. In JAMES II we already have in-memory and disk-based solutions. They adopt the write-and-read-back approach.

Stream data processing has shown to be an effective solution in many areas that require long-running queries over continuous unbounded streams of data (Arasu, Babcock, Babu, Cieslewicz, Ito, Motwani, Srivastava, and Widom 2004), however so far little work exist in the area of simulation. Stream data processing means to process data on the fly and thus implies iterative algorithms and a minimum of storage operations (Chakravarthy and Jiang 2009). Based on JAMES II (Himmelspach and Uhrmacher 2007) we realize a new data storage plug-in which can be used alternatively to the already realized solutions.

1.1 Architecture of the Streaming Data Management Plug-in

Figure 1 shows the constructing entities of the streaming data management plug-in which utilizes streams for input handling and functions for processing. Named streams forward incoming data to functions which have been registered for the corresponding trajectories. Functions evaluate successively on forwarded data thereby updating their internal state. Both streams and functions buffer their data. Here, buffering enables (1.) subsequent functions to read back former data, (2.) the simulation system to read back managed and evaluated data, and (3.) the streams and functions to perform more efficient bulk transfers to the data store if their data is selected to be stored persistently. An exchangeable buffering strategy (not shown here) is responsible for flushing and adjusting the buffers individually. The context of a stream is determined by referring to the overall experiment, the configuration, and the replication in JAMES II.

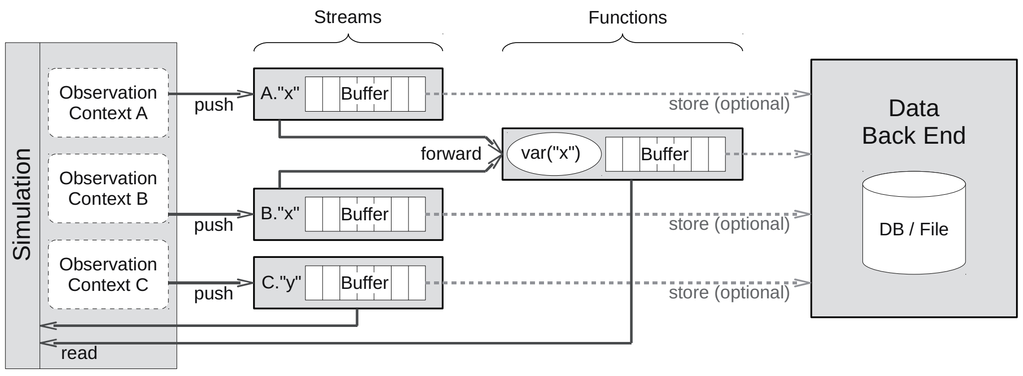

Figure 1: Architecture of the streaming data management plug-in.

1.2 Experimental evaluation

A first experimental evaluation was done after the plug-in had been implemented to see whether it performs as well as the in-memory solution already available. For this, 30,000 stochastical replications of a species reactions model have been computed. At the end of each replication the model state (vector of 7 integers) was processed by the data storage alternatives in order to compute the variance of each vector element (as part of simulation control). The memory data storage simply stores the model state in internal tables to be read back later by the simulation system which calculates the variances. The stream-based data storage manages the data as described, i.e. it iteratively calculates the variances as soon as the trajectory data arrives and holds the results ready for simulation control.

In our experiments we obtained 11 % shorter run times using the stream data storage compared to using the in-memory data storage.

2 OUTLOOK

Further experiments will be done to show how expensive data processing is if even more data needs to be read back from the in-memory data storage in comparison to the stream-based solution. Also comparisons with the disk-based alternatives are planned. Therefore, the streaming will be parameterized so that it stores data persistently. Thereby, also insights into the performance of the different alternatives in handling large scale experiments will be achieved.

REFERENCES

Arasu, A., B. Babcock, S. Babu, J. Cieslewicz, K. Ito, R. Motwani, U. Srivastava, and J. Widom. 2004. "STREAM : The Stanford Data Stream Management System". Technical report, Stanford InfoLab.

Chakravarthy, S., and Q. Jiang. 2009. *Stream Data Processing: A Quality of Service Perspective*. New York: Springer Science+Business Media.

Himmelspach, J., R. Ewald, S. Leye, and A. M. Uhrmacher. 2010, September. "Enhancing the Scalability of Simulations by Embracing Multiple Levels of Parallelization". In *Proceedings of the 2010 International Workshop on High Performance Computational Systems Biology*: IEEE CPS.

Himmelspach, J., and A. M. Uhrmacher. 2007, March. "Plug'n simulate". In *Proceedings of the Spring Simulation Multiconference*, 137–143: IEEE Computer Society.