An InfoSphere Streams Based Approach for Implementing an FX-Style Auto-Correlation Spectrometer with Accelerator Support

Mahmoud S. Mahmoud[1], Andrew Ensor[1], Alain Biem[2], Bruce Elmegreen[2] and Sergei Gulyaev[1]

[1] Institute for Radio Astronomy & Space Research (IRASR), AUT University, NZ. [2] IBM Thomas J. Watson Research Center, Yorktown Heights, NY.

InfoSphere Streams & the SKA

The phased approach to SKA construction and Exascale data processing/transport requirements demand computing technology that combines vast data processing with high scalability. InfoSphere Streams is a data stream management system (DSMS) middleware designed to ingest, filter, analyze and correlate enormous amounts of data incoming from an unlimited number of data stream sources. Streams' intention is to enable organizations and institutions to rapidly respond to their respective changing environments without the need to store and later process huge amounts of data. Streams aims to fulfil its intention by achieving the following objectives [1]:

- The ability to scale up or cluster to a wide variety of hardware architectures as the demand for more processing power increases.
- Provide automation for the handling of data streams that is responsive to changing user requirements as well as data and system resource availability.
- Incremental tasking for changing data schemes and types.
- Secure transmission of data streams at all system levels, along with comprehensive auditing of the execution environment.

Streams is designed to be highly scalable, so it can be deployed on a single or thousands of computing nodes that may vary with respect to their hardware architecture. The runtime environment or Stream Processing Core (SPC) executes numerous long running queries, which Streams refers to as jobs [2]. The jobs (or long running queries) are represented by a dataflow graph, where each vertex in the graph is referred to as a Processing Element (PE) and each connecting edge is in actual fact a data stream.

Streams Processing Application Declarative Engine (SPADE)

Constructing a distributed stream processing application is a complex process. Essentially the programmer must make the following considerations:

- The data stream transform operations which must be developed. Each set of transform operations is a building block in the chain of transformations that ingest, process, analyze and produce the desired output data stream.
- How the data stream transform operations should be deployed to a distributed computing environment to maximize efficiency so that results are produced accurately and in a timely manner.
- The interconnections, network protocols, scheduling and synchronization of operations between the available computing nodes.

SPADE is designed to conceal the aforementioned considerations thus allowing the programmer to focus solely on the design of a distributed stream processing application. SPADE fulfils its design objectives by collaborating with the SPC to provide a dynamic runtime code generation framework capable of achieving scalability and performance through automatic deployment and optimization [3]. The SPADE language contains generalized relational stream operators. The stream operators are used to construct relational long-running queries. UDOPs (User Defined OPerators) are used for specific situations that the generalized operators cannot fulfil. The UDOPs themselves can be written in Java or C++.

SPADE Application Dataflow Graph

The following SPADE application represented by the dataflow graph of its respective data-stream connected PEs performs a frequency domain conversion followed by an auto-correlation and integration to produce the power spectral density of the signal detected by a single antenna. The application makes use of computing nodes that have access to accelerators such as Cell BEs, GPUs, FPGAs, etc. for performing the frequency domain conversions.

UDOP (User Defined OPerator) PEs are used to implement the accelerator device specific logic on the respective host. The accelerator devices are used to perform the time to frequency domain conversion. Communication between host and accelerator device is performed asynchronously in combination with multi-buffering.

This **FUNCTOR** PE receives complex number value frequency domain data and performs the auto-correlation computation. This steps reduces the data and produces real numbers values.

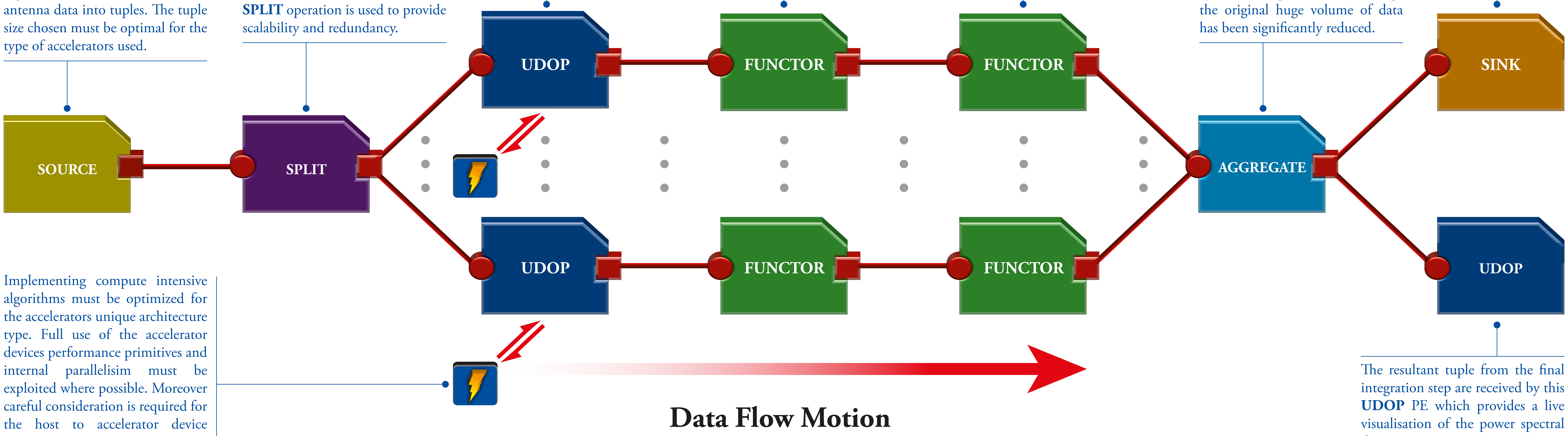
In most cases the initial size chosen for a tuple to be optimal for a particular type of accelerator device usually implies that it contains data values for many frequency domain operations. Hence this **FUNCTOR** PE performs an integration operation to aggregate the auto-correlated results for the frequency domain operations performed within a tuple.

The significantly reduced data is persisted to a suitable store by this **SINK** PE. The persisted data can be used to perform longer integrations or deeper spectral analysis.

The **SOURCE** PE is responsible for ingesting, parsing and collating digitized real value time domain antenna data into tuples. The tuple size chosen must be optimal for the type of accelerators used.

The **SPLIT** PE distributes tuples to the PEs with underlying accelerator device enabled hosts. Effectively the **SPLIT** operation is used to provide scalability and redundancy.

The **AGGREGATE** PE performs a summation and averaging for a specified integration. At this stage the original huge volume of data has been significantly reduced.



Implementing compute intensive algorithms must be optimized for the accelerators unique architecture type. Full use of the accelerator devices performance primitives and internal parallelisim must be exploited where possible. Moreover careful consideration is required for the host to accelerator device communication link.

The resultant tuple from the final integration step are received by this **UDOP** PE which provides a live visualisation of the power spectral density.

Accelerators

The SPADE auto-correlation spectrometer application has been successfully tested using Cell BE accelerators in observations made with the AUT 12m radio telescope at Warkworth, NZ. Current on going research is focussed on increasing the host to accelerator device communication bandwidth as well as using NVIDIA GPU based accelerators. InfoSphere Stream's flexibility to accommodate various hardware architectures provides a platform for carrying out extensive comparison analysis between various hardware architectures. Comparison of various commodity computing architectures will develop a better understanding towards which types of accelerators are best suited for particular stream-data pipeline operations.

References

[1] Bollard, C., Farrell, D. M, Lee, M., Stone, P. D., Thibault, S., & Tucker, S., (2010). IBM InfoSphere Streams: Harnessing Data in Motion. IBM Redbooks: International Business Machines Corporation.

[2] IBM Corporation (2009). System Infrastructure for Streaming. Retrieved from http://domino.research.ibm.com/comm/research_projects.nsf/pages/esps.Projects.html

[3] Gedik, B., Andrade, H., Wu, K., Yu, P. S., & Doo, M., (2008). SPADE: the system S declarative stream processing engine. Proceedings of the ACM SIGMOD international conference on Management of data. DOI: 10.1145/1376616.1376729

IBM AUT UNIVERSITY TE WĀNANGA ARONUI O TAMAKI MAKAU RAU IRASR