The SRC Advantage for Open Source Software Frameworks

SRC Computers, LLC
www.srccomputers.com

OVERVIEW

An extraordinary number of open source software frameworks are freely available today. These frameworks may provide a certain computation methodology, enhanced system wide functionality, or simply make problem solving easier for an application developer. There are many different types of software frameworks including: streaming computation (Apache Storm), map-reduce computation (Apache Hadoop), distributed grid processing (BOINC), multimedia processing (GStreamer), and heterogeneous system middleware (OpenCPI), to mention a few. SRC offers developers who utilize these frameworks the same advantages available to any application deployed on SRC heterogeneous reconfigurable systems: higher application performance, lower power consumption, reduced footprint, and superior total cost of ownership.

“Framework” in this discussion means an execution environment wherein application functionality is defined by the programmer, but exactly how the application executes is determined by the framework. For example, in Apache Hadoop, the programmer defines application computation by writing Map and Reduce style functions while Hadoop provides the control flow that calls these Map and Reduce functions.

CARTE++ IS THE KEY

All SRC systems contain at least two processors: one general purpose microprocessor and one FPGA-based MAP® processor (IMPLICIT+EXPLICIT Architecture™). The SRC-developed Carte++ Programming Environment® compiles C/C++ code for the microprocessor and MAP processor into a unified executable, which may be a single file or a collection of functions in a shared object library. The Carte++ run-time environment automatically loads an FPGA bitstream from a unified executable whenever a MAP processor function call is encountered in the program.

This combination of creating processor-unified executables and the system run-time environment enables Carte++ programs to work well with existing software frameworks. In general terms, a given framework runs on an SRC system microprocessor under Linux. Functions invoked under the framework execute on one or more MAP processors, in cooperation with zero or more system microprocessors as determined by the application programmer.

APACHE STORM

SRC systems and Carte++ natively support Apache Storm, a streaming computation framework. The Carte++ programming model already embodies streaming processing among parallel sections within a program. Storm may be viewed as a framework that naturally extends Carte++ streaming across multiple system components. In the Storm framework, “Spouts” produce data and “Bolts” process data. Spouts and Bolts are usually spread across many TCP/IP networked computer systems. One advantage SRC systems offer Storm developers is a significant reduction in the number of Bolts required for an application due to the MAP processor’s computational efficiency. A good rule of thumb is one MAP processor replaces at least one hundred microprocessors. This Bolt processor compaction effect leads to greater overall performance due to MAP processor efficiency and reduced TCP/IP traffic in a Storm network.
APACHE HADOOP

SRC systems and Carte++ natively support Apache Hadoop, an execution framework in which a Big Data application is defined by a “Map” and a “Reduce” function connected together by a “Shuffle” function. The Map function performs an operation on all input elements, the Shuffle function groups Map result data while the Reduce function performs an operation on each group of shuffled data. Similar to the Storm framework, MAP processor efficiency reduces the number of required application processors and inter-processor communication, thereby increasing execution performance.

BOINCG GRID COMPUTING FRAMEWORK

SRCs systems provide outstanding performance for signal processing, molecular biology, system simulations and many other scientific applications. The University of California at Berkeley created the BOINC grid computing framework, establishing a cooperative distributed computing framework for dozens of scientific applications. These include SETI@home, a large-scale radio signal processing application; Rosetta@home, which studies biochemical functionality in cellular systems via protein folding; and LHC@home, which runs distributed simulations to improve the design of the Large Hadron Collider at CERN. Adding the computational power of SRC heterogeneous reconfigurable computers to a BOINC scientific project is easy using the BOINC client application programming interface (API) and Carte++. The developer only needs to add required BOINC client API functions, a work unit data parser, and a result file generator to their SRC scientific application.

GSTREAMER

SRC systems achieve superior performance for audio and video processing applications. GStreamer is a modular multimedia processing framework. Streaming audio and video data are processed in a Gstreamer pipeline. A pipeline consists of functional “plug-ins” connected together with each plug-in contributing to the overall multimedia processing. A plug-in is simply a Gstreamer-wrapped function call and all reside in one or more Linux shared object libraries. Which plug-ins are used from a library and how they are connected together in the pipeline is defined at Gstreamer application start-up time. The Gstreamer framework provides an easy method to configure and run multimedia processing modules on SRC systems.

OPENCPI

OpenCPI is a middleware framework that enables application execution on inter-connected heterogeneous processing elements. The elements may be general purpose microprocessors, FPGAs, and GPUs. SRC already provides peer execution among its closely coupled microprocessors and FPGA based MAP processors, so OpenCPI provides no additional advantage on SRC systems alone. However, OpenCPI is very useful if an application developer requires a framework to couple SRC systems with other heterogeneous processing elements such as GPUs. Note that the microprocessor and MAP processor in an SRC system are defined as autonomous general purpose processors within a larger OpenCPI framework.

SUMMARY

There are far too many open source software frameworks to attempt to describe them all here. At the same time, the answer to the question of whether or not a framework will run on SRC systems boils down to only two questions:

1. Does the framework run under Linux?
2. Does the framework have the ability to do either of:
   a. Call compiled C functions from a Linux shared object library?
   b. Invoke executable programs under Linux?

If the answer to both of these questions is “yes”, then SRC heterogeneous reconfigurable systems with the Carte++ Programming Environment will support the framework in question. It’s really that simple.

1 SRC Application Implementations on a Saturn 1 Processor, SRC Computers LLC, MKT-054-00, August 3, 2012.