FifthGen White Paper

The SPMD Operation Mode on a Binary Tree Computer System

September 26, 2011

Prepared by
Fifth Generation Computer Corporation
445 Park Avenue, 9th Floor
New York, NY 10022

The SPMD Operation Mode on a Binary Tree Computer System

September 26, 2011

Thomas O. Jones

In a parallel processing system configured as a binary tree computer system, the major application programming construct is “divide-and-conquer” in which the identical program is executed simultaneously in each Processing Element (PE) but on different data sets. The multiple executions of this single procedure can follow different instruction streams (still consistent within the procedure) depending on the data. These potentially different instruction streams are forced to converge and synchronize at the completion of the individual procedures. This synchronization scheme is known as barrier synchronization.

This mode of operation is referred to as SPMD**,** an acronym that stands for Same Program Multiple Data.

In support of the SPMD operation mode, several global burst communications are invoked by function calls from the host, and are examples of traffic-specific interprocessor communications. *Broadcast* transmits data from the host to the PEs. *Resolve* identifies the PE in the tree with the maximum value of some variable, such as a probability score. The *Report* command, typically following a Resolve, transmits data from the identified PE to the Host computer, such as the name of the reference pattern that has the maximum probability of matching an unknown pattern.

AllReduce is a combination of the *Resolve* and *Report* commands.

Background

The DADO machine was developed in the Computer Science Department at Columbia University during the period from 1981 through 1989. DADO is a massively parallel processor (MPP) comprised of an array of processing elements (PEs) *interconnected in a binary tree configuration*. Patents related to DADO and the copyrighted software programs were transferred to Fifth Generation Computer Corporation (FifthGen) in a 1996 agreement.

FifthGen’s patented Binary Tree Computer System utilizes a light weight single threaded kernel operating system to control the operation of the compute nodes.

The DADO Kernel and BootLoader Programs

In computer science, the **kernel** is that piece of software responsible for providing secure access to the machine's hardware to other computer programs. Accessing the hardware directly can be very complex, so kernels usually implement some hardware abstractions to hide complexity and provide a uniform interface to the underlying hardware, thereby simplifying the task for application programmers

The FifthGen DADO C-based Kernel Source Code Listing was written during the period 1982 through 1985 and registered with the Library of Congress Copyright Office in 1994. The Program listings were clearly marked with a copyright notice.

The Fifthgen DADO Kernel was summarized by its authors at that time as follows:

"The lowest level of the kernel actually manipulates logic levels; an intermediate level implements protocols with these signals. The highest level is accessible to the user, and provides the following basic functions:

1. **Broadcast** to send information to the descendant processors.
2. **Resolve** to select one processor from a candidate set (to determine MAX or MIN).
3. **Report** to send information to the root processor (this is termed, "**Allreduce**" in MPI).
4. **Send** to send data to a neighbor.
5. **Receive** to receive data from a neighbor.
6. **Enable and Disable** to make processors execute the instructions or to prevent them from doing anything except forward data.
7. **MIMD and Synch** to partition the tree into independently executing subtrees, and to resynchronize."

The DADO Machine Defined the SPMD Mode of Operation for Parallel Processing Systems

Sometime in 1983, the first implementation of the SPMD Mode of Operation in a Parallel Processing System occurred at Columbia University during the early development of the DADO Parallel Computer.

Daniel P. Miranker, a leading member of the development team, published an article, "Performance Analysis of Two Competing DADO PE Designs, dated November 15, 1983," (See Exhibit C) in which he reported:

> "SIMD parallelism is typically controlled by a conventional processor. The control processor issues a stream of machine level instructions that are executed synchronously in lock step by all the slave processors in the array. DADO is different. Since each PE of DADO is a fully capable computer, and communication between PE's is generally expensive, we wish to make an instruction as "meaningful" as possible. What is communicated as an instruction in DADO is actually a pointer to a procedure, stored locally in each slave PE. Primitive SIMD DADO instructions are in fact parallel procedure calls and may be viewed as macro instructions.
>
> "For example, a common instruction that will be executed by a DADO PE is "MATCH (pattern)," where MATCH is a generalized pattern match routine local to each processor.''

First Published Use of the term SPMD (Single Program, Multiple Data Stream)

In the January 1987 issue of COMPUTER magazine, Professor Salvatore J. Stolfo, the leader of the DADO project (which was partially funded by DARPA) described some early results in an article entitled, "Initial Performance of the DADO2 Prototype," attached as Exhibit D to this report.

> "On December 5, 1985 a 1023-processor parallel machine named DADO2 was successfully demonstrated at Columbia University. DADO2 is the fourth prototype, but the first large-scale prototype of a class of machines called DADO.
>
> "DADO was first proposed in 1980 as a special-purpose parallel computer attached to a conventional host processor and designed to accelerate a particular class of artificial intelligence rule-based programming paradigms called *production systems.*

Later in the same article, Professor Stolfo elaborated on the SPMD mode of operation:

> "DADO2 provides parallel remote procedure invocation in the style of SIMD processing. The procedures are stored locally within the PEs, operate autonomously, and, therefore, may take different amounts of time to complete. Machine level instructions are not broadcast and executed in lock-step. This mode of operation may be regarded as SPMD (for single program, multiple data stream) execution.

In 2007, an article was published as part of the 2007 IEEE International Conference on Cluster Computing entitled, "Efficient Offloading of Collective Communications in Large-Scale Systems," by Sancho, Kerbyson and Barker. In the introduction, the authors stated the following:

> "The single program multiple data (SPMD) programming model is generally the preferred programming model in parallel scientific applications as they usually exhibit a rich degree of data parallelism."