A New Style of Parallel Computing

Dr. James A. Brown; NestedComputing Corp; SmartArrays, Inc.
Peter Schade; International Microsystems

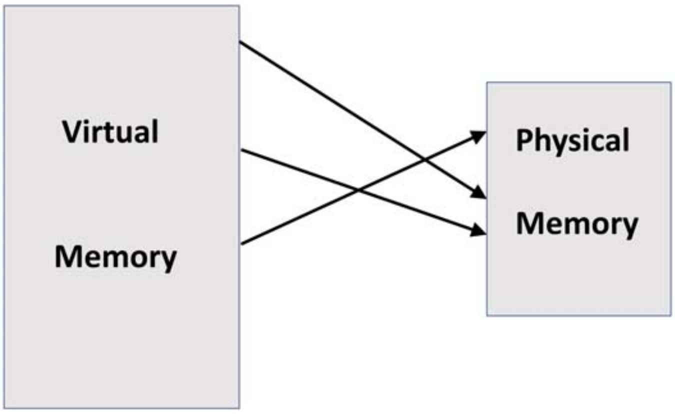
German Guide Share Europe Working Group, APL Germany and IBM Germany
April 16-17, 2018 ERGO Group; Köln, Germany

Abstract

This paper is extracted from the HTML presentation made at the conference. Without the discussion that accompanied the presentation, not everything is easily understood. However, some text is added below for clarification. This is an update to the presentation made at German Guide Share Europe Working Group, APL Germany and IBM Germany November 27-29, 2016 IBM Böblingen, Germany. For some material that has not changed significantly, you are referred to this document: "APL Journal" 1-2/2016.

1. Motivation - Processors

a. Lots of processors on a chip
   i. My laptop has 8 cores
   ii. Intel 48-Core "Single-Chip Cloud Computer
   iii. Adapteva Epiphany architecture allows 2048 processors on a chip
b. All share the same main memory
c. Memory Management Unit – MMU
   i. Translates virtual address to physical address



   ii. All memory references pass through it

d. Memory contention limits performance
   i. levels of cache memory - store information the CPU is most likely to need next
   ii. sophisticated algorithms - it gets complicated

2. **What we’re Doing – These are acronyms but won’t be defined just yet**

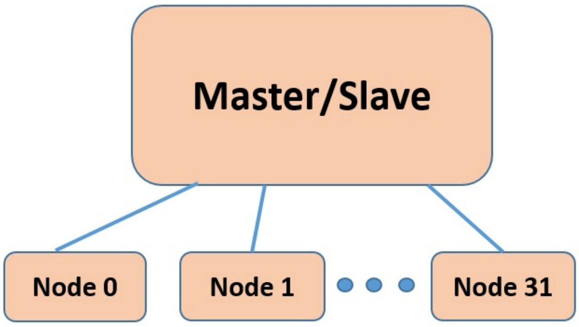
   a. DAC (Patent my International Microsystems)

   b. FASDAC (Copyright 2016 NestedComputing, SmartArrays)

3. **DAC**

   a. The DAC Computer

      i. In it’s most general definition, DAC is a three level machine. Only a two-level machine is considered here.



      ii. This looks generic

      iii. Each box is an general purpose computer (Intel 64-bit)

      iv. It's the connecting lines that are different

   b. Memory model

      i. Attached Memories: (disk drives)

         1. - - Sometimes attached memory is local (on the Master/Slave)
         2. - - Sometimes attached memory is remote (on the nodes)

      ii. An Array can be stored on attached memories:

         1. - - Parts of array stored in multiple separate memories
         2. - - If Master owns memories, the array is local
         3. - - If Nodes own memories, the array is distributed

      iii. We need a new word for an array that is sometimes local and sometimes distributed

      iv. We call such an array a “Disjoint Array”

      v. A computer that supports Disjoint arrays is called a “Disjoint Array Computer”

      vi. This **DAC**

   c. Memory connections

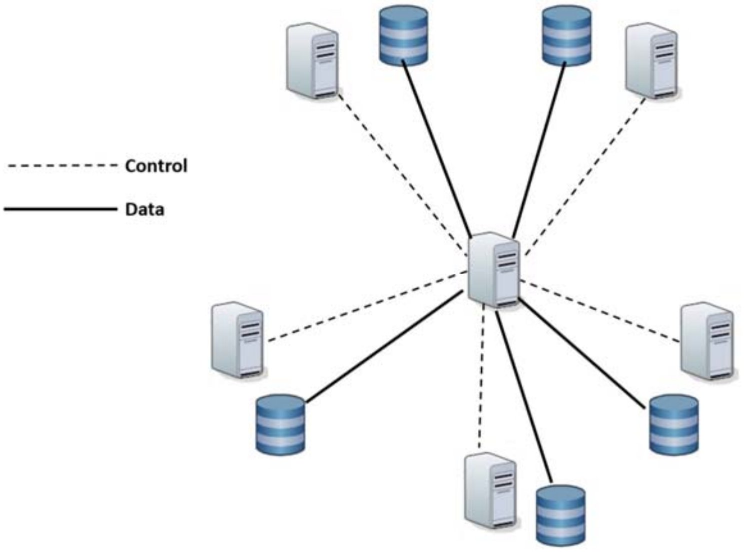
      i. Control Line – – – – – – – – – –

         1. - - Slow Speed Hardware Connection
         2. - - A private internet connection
         3. - - For commands, return codes, messages

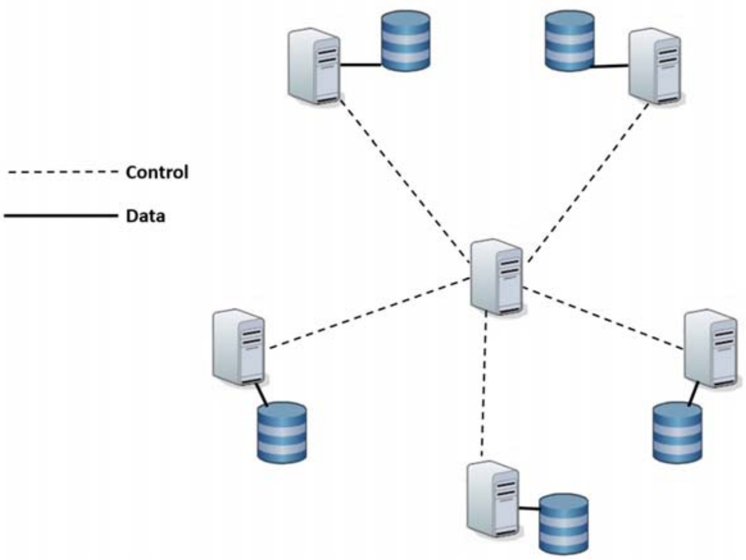
      ii. Data Line ━━━━━━━━━━

         1. - - High Speed Hardware Connection
         2. - - For high volume data

4. **DAC Memory Model – two views**
    a. View 1: DAC Disks attached to Master/Slave



    b. View 2: DAC Disks attached to the Nodes

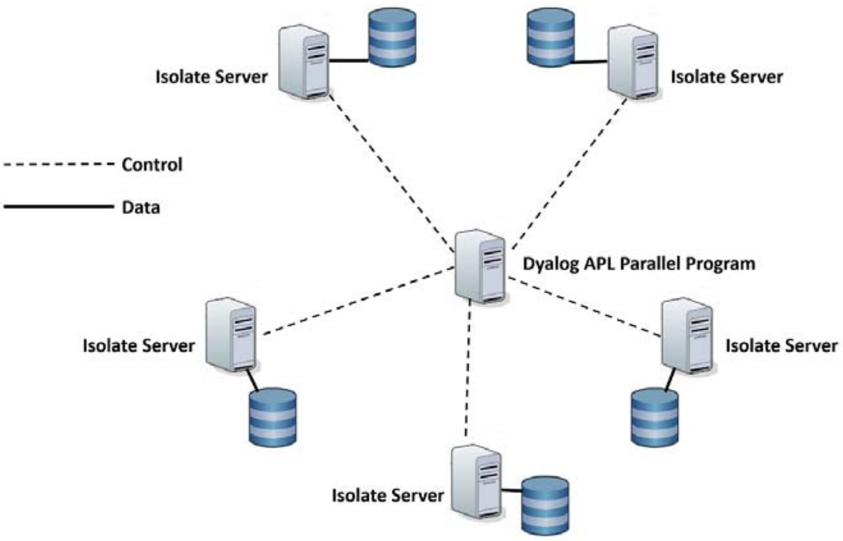


    c. Hardware switch between Master and Node determines where disk is attached

5. Several prototypes have been built
   You are referred to the 2016 paper for pictures and discussion of the prototypes.

6. The programming interfaces have not changed from 2016

   You are referred to the 2016 paper for discussion of the programming interfaces.
   However, of special interest is the ability to use Dyalog APL Isolates as a means for parallel programming on a DAC machine:



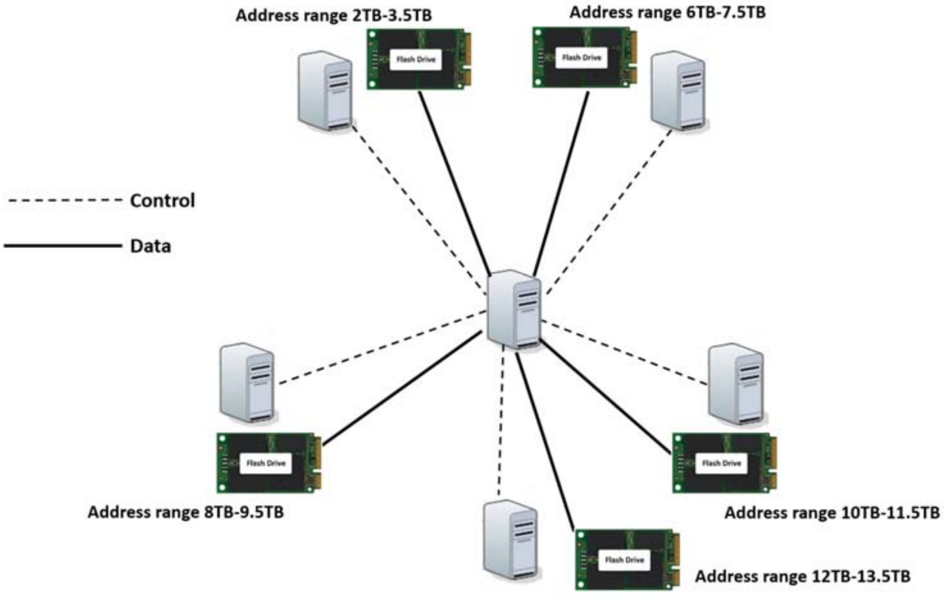
7. FASDAC - Flat Address Space DAC

a. Attached disks treated as RAM
   i. In one or two generations, attached disks will be as fast as RAM
   ii. Plug and play RAM
   iii. All data is memory resident
   iv. Computer has multiple ranges of Random access memory

8. FASDAC Memory Model – two views

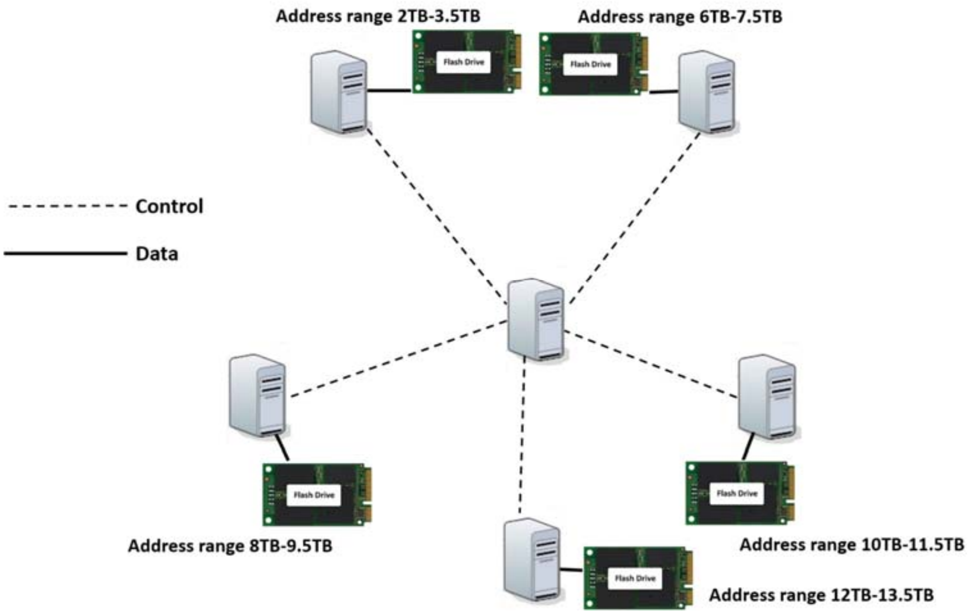
a. View 1: FASDAC Disks attached to Master/Slave

Slave has it's own RAM address space + five attached address spaces



b. View 2: FASDAC Disks attached to the Nodes

Each node has it's own RAM address space + an attached address space



9. Typed Memory

a. Memory is "Typed"
   i. not just a stream of bytes
   ii. Array Oriented
b. Some memory is strongly typed
   i. Type RAM - a stream of bytes for support of legacy software and program execution
   ii. The machine's native memory is type RAM
   iii. Type FILE - memory that contains the contents of a file
   iv. Type OSxx - set of memory items formatted for Operating System control
c. Some memory is weakly typed
   i. Type NUMERIC - a stream of numbers without respect to storage format
   ii. Type CHAR - a stream of characters without respect to storage format (UCS 1,2,4)
   iii. Machine supports conversion
d. Some memory is recursively typed
   i. Type NESTED - memory that contains other memory entries
e. Attached DAC drive
   i. Is recursively typed
   ii. Nested structure matches directory structure
f. Database table
   i. List of columns
   ii. Each column in strongly typed memory
   iii. Trivial memory map for analytics
g. Overallocation for transactions

10. **FASDAC is An Array Machine**
    a. The machine understands Arrays
        i. - - Hardware recognizes linear collections
        ii. - - Higher Rank arrays realized by software
    b. The Operating System understands the hardware and Arrays
        i. - - Array allocation of memory
        ii. - - Management of disjoint arrays
        iii. - - Management of multiple RAM Address ranges
        iv. - - It's an "Array Operating System"
    c. Enabling technologies
        i. - - Hardware: Flash and solid state memory
        ii. - - Software: APL and it's derivatives
        iii. - - It's an "Array Machine"

11. **DAC FASDAC Brief Summary**
    a. Imagine a traditional computer with 50 processors
        i. Numbered 1 through 50
        ii. Each processor processes 2% of the data
        iii. All memory references go through same memory channel – MMU
        iv. Memory contention limits performance
    b. Imagine a 51 Processor DAC or FASDAC Computer
        i. Numbered 0 through 50
        ii. Processor 0 (Master/Slave in first view)
            1. - - Owns all the storage where data resides
        iii. Processors 1 through 50 (Nodes in second view)
            1. - - Each owns 2% of storage where data resides
            2. - - Each has its own path to memory
            3. - - Each has its own Memory Management Unit (MMU)
        iv. In both views, it’s the same memory
    c. Hardware Switches
        i. One switch between Master and each Processors 1 to 50
        ii. Switch for processor 17 in position 1
            1. - - Processor 0 owns that portion of the storage
        iii. Switch for processor 17 in position 2
            1. - - Processor 17 owns 2% of storage

12. Benefit of DAC FASDAC
    a. Has the effect of moving data from Master to Nodes but no data really moves
    b. At compute time, 50 Paths to memory instead of one
    c. Zero memory contention - not minimal contention - zero!

13. Benchmarks
    You are referred to the 2016 paper for pictures and discussion of the benchmarks.

14. Conclusion

DAC and FASDAC solve the memory contention problem for parallel computation

APL and its derivatives are the driving force