EasyChair Preprint

№ 11001

Computer Architecture and Algorithms for High Performance Computing Through Parallel and Distributed Processing

Venkata Subba Reddy Poli

October 1, 2023

Computer Architecture and Algorithms for High Performance Computing through Parallel and Distributed Processing

P. Venkata Subba Reddy

Abstract—There is a very high need of High Performance Computing (HPC) in Many applications like space science to Artificial Intelligence. HPC shall be achieved through Parallel and Distributed Computing. In this paper, Parallel and Distributed algorithms are discussed based on Parallel and Distributed Processors to achieve HPC. The Programming concepts like threads, fork, sockets and par do are discussed with some simple examples for HPC.

Index Terms— High Performance Computing, Parallel and Distributed Algorithms, Computer Architecture, Computer Programming

1 INTRODUCTION

Computer Architecture and Programming play an important role for High Performance computing (HPC) in large applications Space science to Artificial Intelligence []. The Algorithms are problem solving procedures and later these algorithms transform in to particular Programming language for HPC. There is need to study algorithms for High Performance Computing. These Algorithms are to be designed to computer in reasonable time to solve large problems like wather forecasting, Tsunami, Remote Sensing, National calamities, Defense, Mineral exploration, Finite-element, Cloud Computing, and Expert Systems ect. The Algorithms are Non-Recursive Algorithms, Recursive Algorithms, Parallel Algorithms and Distributed Algorithms.

The Algorithms must be supported the Computer Architecture. The Computer Architecture is characterized with Flynn's[2] Classification SISD, SIMD, MIMD, and MISD. Most of the Computer Architectures are supported with SIMD (Single Instruction Multiple Data Streams). The class of Computer Architecture is VLSI Processor, Multiprocessor, Vector Processor and Multiple Processor[1,3].

2 ALGORITHMS

There are Non-Recursive Algorithms, Recursive Algorithms, Parallel Algorithms and Distributed Algorithms[5,10].
In the following Algorithms, Computer Programming and Architectures are discussed

2.1 Non-Recursive Algorithms

Non-Recursive Algorithms are systematically applied to the problems by analyzing the efficiency.
Consider the algorithm of finding maximum element in the array A ([0 .. n-1])

```
maxval ← A[0]
for i←0 to n-1 do
    if A[i] > maxval
        axval ← A[i]
return maxval
```

The problem will be analyzed as

C(n) is number of computations. Where n is input size. The number of times of operations in execution

$$C(n)= \sum_{i=1}^{n-1} 1 = n=1 \Theta(n)$$

2.2 Recursive Algorithms

The recursive algorithm is binary expansion whose number of executions is multiplication.
For instance n! = n*n-1*n-2*…*1
Consider the algorithm of finding n!

If n=0 return 1
Else return F(n-1)*n

The number of multiplications M(n) needs to compute
M(n)=M(n-1) +1 for n>0
M(0)=0

2.3 Parallel Algorithms

Parallel Algorithms are designed to apply on problem to compute parallel whose number of executions is independent.
For instance parallel sum of odd and even number up to A[n].
Consider the algorithm to find parallel sum of odd and even numbers unto A[n].

```
odsum←0, evensun ← 0
for i←0 to n-1 stem 2 do
   oddsum ←oddsum +A[i}]
```

```
return oddsum
for  i←1 to  n-1 stem 2 do
   evensum ←evensum +A[i}]
return evensum
```

Consider the algorithm to find parallel matrix multiplication of A[n, n] and B[n,n].

```
for i←0 to n-1 do in parallel
    for j←0 to n-1 do in parallel
        C[I,j]←0
         for k←0 to n-1 do in parallel
             C[I,j] ← C[I,j] + A[I,k] * B[k,j]
return C
```

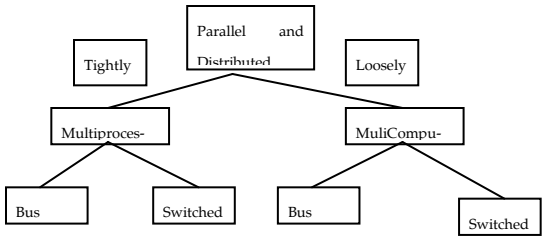
2.4 Distributed Algorithms

Distributed Algorithms are designed to apply on problems to compute on Distributed Systems whose executions are independent and distributed .
Consider the algorithm to compute two transactions
The algorithm of the problems is designed to execute on Distributed Systems.

```
    if (fork())
    {
    #Transaction1;
}
    else
    {
    #Transaction2
}
```

3 PARALLEL AND DISTRIBUTED ARCHITECTURE AND PROGRAMMING

Parallel and distributed Computer Architecture of processors is defined through Flynn's classification (SISD, SIMD, MIMD, MISD)

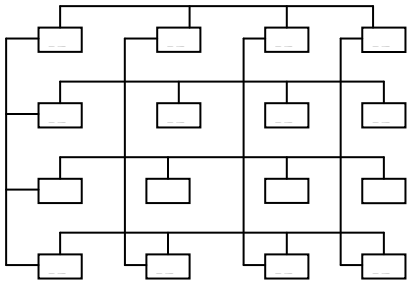


Parallel and Distributed Computer Systems

There are four minimum number of Architecture are VLSI Processor, Multiprocessor, Multiple Processor(Multi Computer) and Vector Processors . These Architectures are discussed in the following

3.1 VLSI Processor

VLSI Chip has Computer components such as Processor arrays (Processing Elements), Memory array and large scale switching networks. Communicate the PEs for implementing Parallel Algorithms with VLSI Chip.



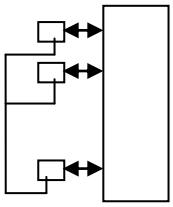
4x4 mesh Processing Elements of VLSI Processor

Consider the Parallel algorithm for odd and even sum of n elements for VLSI Processor.
Compute oddsum, evensum in parallel
One of the PEs set to oddsum:
The Perl program for above parallel algorithm using threads is given by
The "use thread" creates one or more threads.

```
use threads;
        $thr1= threads->new(\&ascending);
        $thr2= threads->new(\&decending);
              ;
         sub ascending {
            my $num=0;
do { $num=$num+1;
 print " $num\n";
         }
          while ( $num<10)
           }
sub decending {
            my $num=10;
do {  print " $num\n";
$num=$num-1;
         }
          while (  $num>0)
}
$thr1-> join;
$thr2->join;
```

3.2 Multi Processor

Multiprocessor Computer Have been modeled as n Processor and Parallel Random Access Machine (PRAM) with shared memory. The Parallel Algorithms will be implemented with PRAM.



Multiprocessor System

Consider the Parallel Algorithm for computation for Multiprocessing System
oddsum←0

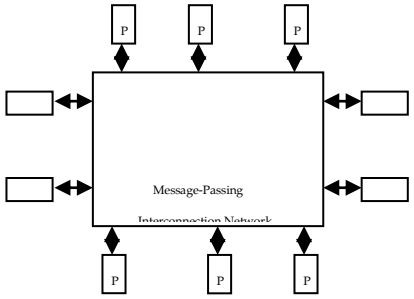
```
for  i←0 to  n-1 stem 2 do
   oddsum ←oddsum +A[i}]
return oddsum

one of the PEs set to evensum:
evensun ← 0
for  i←1 to  n-1 stem 2 do
   evensum ←evensum +A[i}]
return oddsum
```

The Perl Program for above problem using fork for Parallel Processing

3.3 Multi Computer System

Multi Computer involves sequence of routers and channels for as number of Computer Systems with Message Passing Interconnection Network. The Parallel Algorithms will be implemented with this Network

.



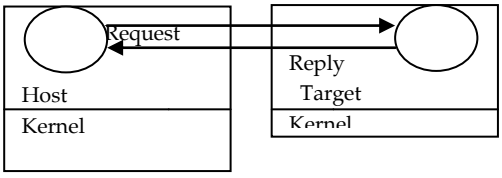
Multi Computer System

Multi computer System/ Distributed system

Consider the Parallel/Distributed algorithms in Multi Computer/distributed System. These algorithms will be computed in two ways Client/Server and Remote Procedural Calls

Remote Procedural Computation

The Client request the Data from the Server and the Server sends the Data from the Server buffer.
The Perl Program gets Time from Server.



Remote Procedural Computation

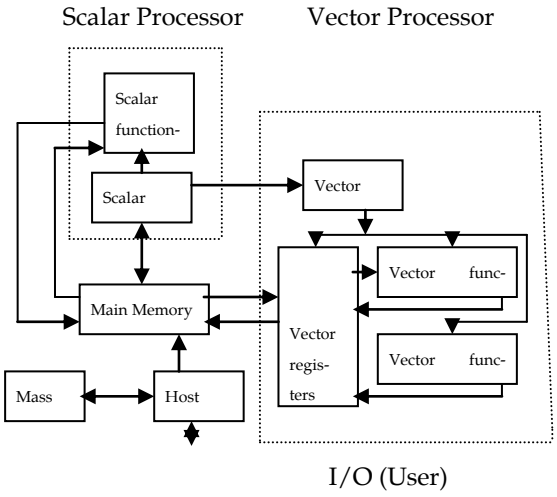
```
#Perl Client  Program
#!/usr/bin/perl
use IO::Socket;
$socket = new IO::Socket::INET (
                        PeerAddr  => '127.0.0.1',
                        PeerPort  =>  7008,
                        Proto => 'tcp',
                        )
or die "Couldn't connect to Server\n";
   $socket->recv($recv_data,1024);
   if($recv_data){
localtime()=$recv_data;
   print "Recieved :$recv_data\n";
   }

#Perl Server Program
#!/usr/bin/perl
use IO::Socket;
$| = 1;
$socket = new IO::Socket::INET (
                        LocalHost => '127.0.0.1',
                        LocalPort => '7008',
                        Proto => 'tcp',
                        Listen => 5,
                        Reuse => 1
                       );
die "Coudn't open socket" unless $socket;
print "\nTCPServer Waiting for client on port 7008";
while(1)
{
        my($new_sock,$buf);
     $buf=sum();
        $client_socket = "";
        $client_socket = $socket->accept();

        $peer_address = $client_socket->peerhost();
        $peer_port = $client_socket->peerport();

        print "\n I got a connection from ( $peer_address
, $peer_port ) ";
     $client_socket->send($buf);
     close $client_socket;
sub sum() { return 2+3;}
}
```

The Host Machine sends the Data to the Target Machines and Target Machine processes the Data and send result Data to the Host Machines.
The Perl program for distributed algorithm may implement using socket & fork.

3.4 Vector Processors

Super Computers are model with Vector Processor. Super Computers are specified by 5-tuples

M = <N, C, I, M, R>
Where
N=number of processors
C= Set of instructions
I is set of instructions for parallel execution
M= Set of Processors
R= Set of routing functions



The Architecture of Vector Supercomputer

Parallel Algorithms are designed to compute big problems like Weather forecasting, Remote sensing, Mineral exploration, Oceanography ect in parallel using Super Computers.

Consider the algorithm to find Parallel Matrix Multiplication A[nxn] and B[nxn], where n is very large.

The Perl program for above parallel algorithm for matix multiplication is given by

for i←0 to n-1 do in parallel
for j←0 to n-1 do in parallel

P[i,j] set to C[i,j]←0
for k←0 to n-1 do in parallel
A[I,k] and B[k,k]broadcast to P[i,j]
C[I,j] ← C[I,j] + A[I,k] * B[k,j]
return C

The Perl program for above parallel programming for matix multiplication using par do in FORTRAN is given by

```
par do 300 i = 1, n
       par do 200 j = 1, n
            par do 100 k = 1, n
      a(i,k) = a(i,k) + b(i,j) * c(j,k)
                  100 continue
             200 continue
   300 continue
```

4 Conclusion

High Performance Computing is required when large computations of the problems. HPC shall be performed through the Parallel and distributed Algorithms. The Parallel and Distributed are discussed based on Computer Architecture. The Class of Algorithms and Class of Computer Architecture are discussed. The Programming concepts like threads, fork, sockets and Par Do are discussed for HPC. Some simple examples are discussed for HPC. The examples shall be extending to large problems like Grid Computing and Cloud Computing. Usually Fotran is used for HPC. The Perl and Java Programming are laso usefull for HPC[11].

Acknowledgment

The author wishes to thank B.Tech., M.Tech., and Ph.D Students for helping me.

References

[1] Kai Hwang, Advanced Computer Architecture, McGraw-Hill, New Delhi, 1993.

[2] M. J. Flynn, "Some Computer Organizations and Theire Effectiveness", IEEE Transactions on Computers, vol.29, n0.9, pp.948-960, 1972.

[3] K. Hwang, Advanced Parallel Processing and SuperCompuer Architecture", Proceedings of IEEE, vol.75, 1987.

[4] K. Hwang anf F. A Briggs, Computer Architecture and Parallel Processing, McGraw-Hill, New Delhi, 1992.

[5] Aho, Hopecroft and Ulman, Design and Analysis of Computer Algorithms, pearson, 2002.

[6] Martin Brown, Perl The Complete Reference, Tata Mc Graw-Hill, New Delhi,2001.

[7] John C. Knight, **The current status of super computers** Original Research Article *Computers & Structures*, *Volume 10, Issues 1–2, Pp.401-409, April1878.*

[8] Horst D Simon Erich Strohmaier, Jack J Dongarra, Hans W Meuer, **The marketplace of high-performance computing** Original Research Article*Parallel Computing*, *Volume 25, Issues 13–14, pp. 1517-1544, Decmber 1999.*

[9] Guillermo L. Taboada, Sabela Ramos, Roberto R. Expósito, Juan Touriño, Ramón Doallo ,**Java in the High-Performance Computing arena: Research, practice and experience** Original Research Article*Science of Computer Programming July 2011.*

[10] N. Sim, D. Konovalov, D. Coomans **High-Performance GRID Computing in Chemoinformatics***Comprehensive Chemometrics*, *pp. 507-539, 2009.*

[11] **P. Venkata Subba Reddy**, "Object-Oriented Software Engineering through Java and Perl", CiiT International Journal of Software Engineering and Technology, July 2010.

Dr. P. Venkata Subba Reddy was Professor and Head, Department of Computer Science and Engineering, MeRITS, Udayagiri, India during 2006-07. He is currently Associate Professor in Department of Computer science and Engineering, College of Engineering, Sri Venkateswara University, Tirpathi, India working since 1992. He did Ph.D in Artificial Intelligence, 1992). Sri Venkateswara University, Tirpathi, India. . He did Post Doctoral/Visiting fellowship in Fuzzy Algorithms under Prof. V. Rajaraman, SERC,IISC/JNCASR, Bangalore, India. He is actively engaged in Teaching and Research work to B.Tech., M.Tech., and Ph.D students. He is actively in doing research in the areas of fuzzy systems, database systems, Software Engineering , Expert Systems and Natural language processing. He published papers in reputed National and International journals. He is an Editor for JCSE