Simulation of Advanced Large-Scale HPC Architectures

Frank Lauer

28. April 2010
Overview

- Motivation
- Design
- Implementation / Results
- Future Aspects
History HPC

Simulation of Advanced Large-Scale HPC Architectures

1 PFlop/s
(10^15)

1 TFlop/s
(10^12)

1 GFlop/s
(10^9)

1 MFlop/s
(10^6)

1 KFlop/s
(10^3)

2X Transistors/Chip
Every 1.5 Years

Super Scalar

Super Scalar/Special Purpose/Parallel

Vector

Parallel


1941 1 (Floating Point operations / second, Flop/s)
1945 100
1949 1,000 (1 KiloFlop/s, KFlop/s)
1951 10,000
1961 100,000
1964 1,000,000 (1 MegaFlop/s, MFlop/s)
1968 10,000,000
1975 100,000,000
1987 1,000,000,000,000 (1 GigaFlop/s, GFlop/s)
1992 10,000,000,000
1993 100,000,000,000
1997 1,000,000,000,000,000 (1 TeraFlop/s, TFlop/s)
2000 10,000,000,000,000
2007 478,000,000,000,000,000 (478 Tflop/s)

UNIVAC 1
EDSAC 1

IBM 7090

CDC 6600

Cray 1

Cray X-MP

ASCI Red

ASCI White

TMC-CM-2

TMC CM-5

Cray T3D

Cray 2

RoadRunner

Crack Jaguar

Frank Lauer
History HPC

Cores Per System

- +128k
- 64k-128k
- 32k-64k
- 16k-32k
- 8k-16k
- 4k-8k
- 2049-4096
- 1025-2048
- 513-1024
- 257-512
- 129-256
- 65-128
- 33-64
- 17-32
- Sep 16
- 05. Aug
- 03. Apr
- 2
- 1
The project aim is to provide a transparent platform for analysing MPI based applications in advanced large HPC architectures.
The project aim is to provide a transparent platform for analysing MPI based applications in advanced large HPC architectures.

- Run unmodified MPI code
The project aim is to provide a transparent platform for analysing MPI based applications in advanced large HPC architectures.

- Run unmodified MPI code
- Scale applications at least to $10^7$ entities
The project aim is to provide a transparent platform for analysing MPI based applications in advanced large HPC architectures.

- Run unmodified MPI code
- Scale applications at least to $10^7$ entities
- Fault injection into the program flow
The project aim is to provide a transparent platform for analysing MPI based applications in advanced large HPC architectures.

- Run unmodified MPI code
- Scale applications at least to $10^7$ entities
- Fault injection into the program flow
- Collecting metric data during a run
Related Work

- BIGSIM
- JCAS
- SST
- $\mu\pi$
- AMPI CHAMP++
- MPI-Sim
Design

- Project based on parallel discrete event simulation (PDES)
Design

- Project based on parallel discrete event simulation (PDES)
• Project based on parallel discrete event simulation (PDES)
Project based on parallel discrete event simulation (PDES)
Design

- Project based on parallel discrete event simulation (PDES)
  - Logical process $\rightarrow$ Virtual compute node
  - Event $\rightarrow$ MPI message
  - Virtual time $\rightarrow$ CPU utilisation time + network delay
Design

- Project based on parallel discrete event simulation (PDES)
  - Logical process $\rightarrow$ Virtual compute node
  - Event $\rightarrow$ MPI message
  - Virtual time $\rightarrow$ CPU utilisation time + network delay
  - Global virtual time
Design

- Project based on parallel discrete event simulation (PDES)
  - Logical process → Virtual compute node
  - Event → MPI message
  - Virtual time → CPU utilisation time + network delay
  - Global virtual time
  - Synchronisations mechanisms
Design

- Project based on parallel discrete event simulation (PDES)
  - Logical process → Virtual compute node
  - Event → MPI message
  - Virtual time → CPU utilisation time + network delay
  - Global virtual time
  - Synchronisations mechanisms

- Reuse of the JCAS architecture
Design

- Project based on parallel discrete event simulation (PDES)
  - Logical process $\rightarrow$ Virtual compute node
  - Event $\rightarrow$ MPI message
  - Virtual time $\rightarrow$ CPU utilisation time + network delay
  - Global virtual time
  - Synchronisations mechanisms

- Reuse of the JCAS architecture
main(int, char**)  
~~~~~~~~~  
~~~~~~~~~  
~~~~~~~~~  
thread_join();  
~~~~~~~~~  
~~~~~~~~~  
~~~~~~~~~  
VM_main(int, char**)
~~~~~~~~~  
~~~~~~~~~  
~~~~~~~~~  
~~~~~~~~~  
some_func(void*)
~~~~~~~~~  
yield();
~~~~~~~~~  

Communication()
{  
  while(comm)  
  {  
    send/recv();  
  }  
}  

Frank Lauer
Simulation of Advanced Large-Scale HPC Architectures

Design - Internal Communication

Graph representation of internal communication pathways between Sim_Consultation, Event Queues, 0, 1, ..., n, and Virtual_Machines.
Design - GVT Synchronisation

GVT

Find smallest element.

LVT
MPI rank 0
Find smallest element.

LVT
MPI rank 1
Find smallest element.

VT
MPI vrank 0

VT
MPI vrank 1

VT
MPI vrank 2

VT
MPI vrank 3

VT
MPI vrank 4
Implementation

- Unmodified code
  - Add header file
  - Compile to object file
  - Link against simulator

- Supported MPI-calls
  - MPI_Init(...)
  - MPI_Finalise()
  - MPI_Comm_rank(...)
  - MPI_Comm_size(...)
  - MPI_Isend(...)
  - MPI_Recv(...)

- Simulation size only restricted by memory
- Variable stack size
Implementation - Known Restrictions/Issues

- Wrapper function calls only in main(...)
- Memory Segmentation fault
- Memory out of Resources
- printf(...) floating point values in 64bit
Test Result - Record Virtual Time

Time accounting for the VT and the GVT synchronisation

Experimental Standard Deviation.

Frank Lauer
Determine the heat distribution on a two dimensional area

\[ T_i = \frac{T_{i-1} + T_{i-1}^{up} + T_{i-1}^{down} + T_{i-1}^{right} + T_{i-1}^{left}}{5} \]
Simulation Test App. one: Heat Transfer

Modified to non blocking finalization.

Virtual Scale Analysis On Modified Application

![Graph showing runtime in seconds vs. virtual comm size for different numbers of MPI nodes (2, 4, and 10).]
Simulation Test App. two: Numerical Quadrature

Solve a integral numerical, by dividing it into subintegrals and using approximation:

\[ S = \sum_{i=0}^{n-1} A_i, \quad A_i = \frac{h_i(f(x_i) + f(x_{i+1}))}{2} \]
Simulation Test App. two: Numerical Quadrature

Modified, where no data is gathered.

Simulation Efficiency At Different Comm Size

- Efficiency vs. MPI Nodes
- Lines represent different communication sizes:
  - 5,000
  - 10,000
  - 50,000
  - 100,000
Future Aspects

- Embed more complex MPI applications
Future Aspects

- Embed more complex MPI applications
- Virtual time for LP
Future Aspects

- Embed more complex MPI applications
- Virtual time for LP
- Enhance MPI instruction set
Future Aspects

- Embed more complex MPI applications
- Virtual time for LP
- Enhance MPI instruction set
- Fault injection
Questions ?
Thank you for your attention!