

Universidade de São Paulo
Instituto de Matemática e Estatística
MAC 5742 - Computação Paralela e Distribuída

Linguagem Chapel

Autor:

Walter Perez Urcia

São Paulo

Junio 2015

Resumo

Nesse Exercício Programa (EP) o objetivo é explorar a computação Distribuída com a biblioteca MPI (Message Passage Interface em inglês). A implementação a usar será OpenMPI. O desafio é entender o uso de MPI para a solução de problemas grandes, os exercícios serão baseados em clássicos programas de soma de reduções, produto escalar e multiplicação de matrizes.

Sumário

1	Introdução	5
2	História	6
2.1	Origem	6
2.2	Versões	6
3	Especificações da linguagem	6
3.1	Instalação e configuração	6
3.2	Data Paralellism	6
3.3	Task Paralellism	6
3.4	Multi Locale Paralellism	6
4	Experimentos e resultados	6
4.1	Configuração para os experimentos	6
4.2	Experimento 1: Soma de matrizes	6
4.2.1	Códigos	6
4.2.2	Resultados	6
4.3	Experimento 2: Produto escalar	6
4.3.1	Códigos	6
4.3.2	Resultados	6
5	Conclusões	6

Lista de Figuras

1 Introdução

Message Passing Interface(MPI) é um padrão para comunicação de dados para aplicações que requerem computações paralelas. O padrão define a estrutura e a funcionalidade de uma serie de rotinas que permitem a comunicação entre os processadores, nós de um cluster, etc.

Neste padrão, uma aplicação é percebida como um ou mais processos que se comunicam mediante o acionamento de funções para o envio e recebimento de mensagens. Os processos podem usar mecanismos de comunicação ponto a ponto ou operações coletivas de comunicação (operações globais).

O Open MPI é uma implementação open source de MPI. Open MPI apresenta as seguintes características:

- Conformidade com MPI-3
- Concurrency e seguridade de threads
- Tolerancia a erros nos processos e redes
- Soporte a diversos tipos de redes
- Soporte de mais de um agendador de trabalhos
- Soporte para diversos sistemas operacionais
- Portavel e mantemivel
- Desenho a base de componentes, APIs documentadas
- Licencia BSD

2 História

2.1 Origem

2.2 Versões

3 Especificações da linguagem

3.1 Instalação e configuração

3.2 Data Parallelism

3.3 Task Parallelism

3.4 Multi Locale Parallelism

4 Experimentos e resultados

4.1 Configuração para os experimentos

4.2 Experimento 1: Soma de matrizes

4.2.1 Códigos

4.2.2 Resultados

4.3 Experimento 2: Produto escalar

4.3.1 Códigos

4.3.2 Resultados

5 Conclusões

Nossas conclusões finais são:

- Com OpenMPI não sempre ter mais processos paralelos é sinônimo de melhor tempo de execução. Isto é devido a que o tempo de comunicação entre os processos pode ter um custo elevado.
- Ao executar programas com varias máquinas o tempo de execução pode ser maior do que o tempo da execução sequencial. É importante ter em consideração os tempos de comunicação entre processos de ambas máquinas.
- Para o envio de dados grandes é aconselhável usar as funções não bloqueantes de OpenMPI. Fazer isto reduz os tempos de espera.