Semana sobre Programação Massivamente Paralela 2017

MC-MP09. Adaptive MPI (AMPI)

Exemplos no Santos Dumont

Laércio Pilla (UFSC)

Celso Mendes (INPE)

Esteban Meneses (ITCR)

Conteúdo

- **EXEMPLOS com AMPI**
  a) **HELLO:** Execução com múltiplos VPs por proc.
  b) **JACOBI:** Stencil 2D com vários VPs por proc.
  c) **MAPPING:** Muda distribuição inicial de VPs
  d) **BIGSTACK:** Altera tamanho de stack num VP
  e) **GLOBALS:** Privatização de variáveis globais
  f) **LOADBAL:** Balanceamento de carga

Formato dos Exemplos

- **Para cada exemplo:**
  - Objetivo, ambiente de execução e saídas obtidas com execuções no S.Dumont
  - Em cada sub-diretório, há também o código-fonte, Makefile e script para o job
- **Pré-requisito:**
  - Antes de testar os exemplos, executar:

source /scratch/app/modulos/intel-psxe-2016.2.062.sh

module load gcc-5

Programa HELLO

- **Objetivo:**
  - Ilustrar o uso de processadores virtuais e sua distribuição padrão pelo sistema
- **Ambiente de Execução:**
  - 2 nós, 2 procs/nó: P=4; VP=8
- **Saída:**
  - Identificação de cada VP mostrando onde este VP está sendo executado, em detalhes, incluindo os hostnames

Programa JACOBI

- **Objetivo:**
  - Ilustrar o uso de sobre-decomposição de domínio num programa AMPI, com VP>P
- **Ambientes de Execução:**
  a) 1 nó, 3 procs: P=3; VP=16 (16 ranks MPI)
  b) 1 nó, 6 procs: P=6; VP=16 (também 16 ranks MPI)
- **Saídas:**
  - Apresentação do tempo de execução e número de ranks usados. Com P=6, o tempo deve ser próximo da metade do tempo com P=3

Programa MAPPING

- **Objetivo:**
  - Ilustrar formas de distribuição inicial dos ranks (threads) pelos processadores utilizados
- **Ambientes de Execução:**
  a) 1 nó, 3 procs: P=3; VP=6; Distr.Cíclica (RR_MAP)
  b) 1 nó, 3 procs: P=3; VP=6; Distr.Blocada (BLOCK_MAP)
- **Saídas:**
  - Apresentação da localização de cada VP nos PEs disponíveis, em acordo com cada distribuição

Programa BIGSTACK

- **Objetivo:**
  - Ilustrar o uso de um stack aumentado para cada thread
- **Ambientes de Execução:**
  a) 1 nó, 2 procs: P=2; VP=4; stack default = 1 MB
  b) 1 nó, 2 procs: P=2; VP=4; stack de 2.5 MB setado com a opçãp *+tcharm_stacksize*
- **Saídas:**
  - Devido ao uso de um array de 2 MB no programa, erro de execução se o stack for menor que 2 MB.

Programa GLOBALS

- **Objetivo:**
  - Ilustrar a privatização automática de variáveis globais através da opção de compilação *swapglobals*
- **Ambientes de Execução:**
  a) 1 nó, 4 procs: P=4; VP=8; SEM *swapglobals*
  b) 1 nó, 4 procs: P=4; VP=8; usa *swapglobals* para privatizar variável global
- **Saídas:**
  - Valores impressos a cada passo de execução mostram que, com VP>P, o valor da variável global só é preservado se for usada a opção de compilação *swapglobals*

Programa LOADBAL

- **Objetivo:**
  - Ilustrar o funcionamento de um balanceador de carga num programa iterativo com AMPI
- **Ambiente de Execução:**
  - 2 nós, 2 procs/nó: P=4; VP=8; balanc.=GreedyLB; carga inicial nos 4 primeiros VP é muito menor que a dos 4 últimos VPs
- **Saídas:**
  - Valores impressos mostrando a localização de cada VP. Informação detalhada sobre as etapas de balanceamento, feitas a cada 100 iterações. Notar que o tempo das 100 primeiras iterações é o dobro do tempo das 100 seguintes, devido ao desbalanceamento inicial existente.