CAP-387(2016) – Tópicos Especiais em Computação Aplicada: Construção de Aplicações Massivamente Paralelas

Aula 4: Desempenho, Modelos e Benchmarks

Celso L. Mendes, Stephan Stephany

LAC /INPE

Emails: celso.mendes@inpe.br, stephan.stephany@inpe.br

Desempenho de Aplicações

- Aplicações Paralelas:
  - Em geral: computação, comunicação, I/O
  - Desempenho global: tempo total de execução
    - Efetivamente, é a métrica básica a ser otimizada
  - Outras métricas de interesse
    - Total de memória utilizada
    - Escalabilidade com o número de processadores usados
  - Porém, deve-se tomar cuidado em alguns casos!
    - Melhor escalabilidade nem sempre significa melhor desempenho!

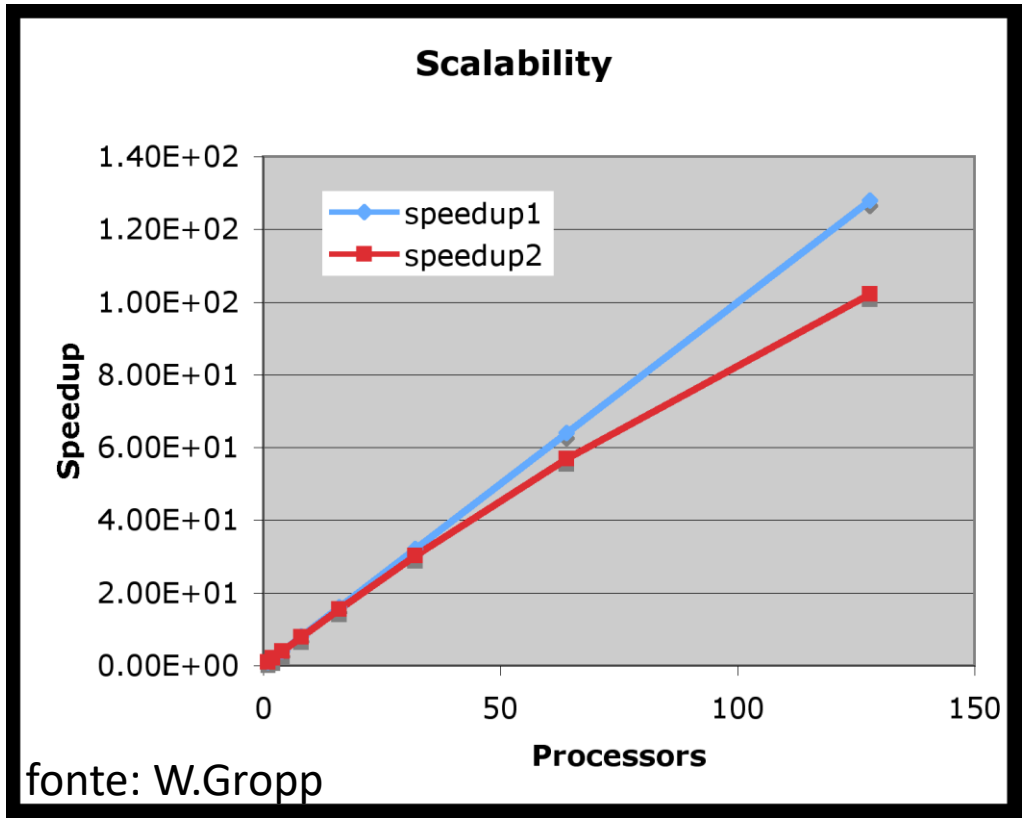
Comparação de Desempenho

- Exemplo: dois algoritmos paralelos, (1) e (2)

  - Suponha que ambos resolvem o mesmo problema, porém com diferentes valores de speedup:

    (1): Speedup quase ideal
    (2): Speedup inferior

    Qual o melhor algoritmo?



fonte: W.Gropp

Comparação de Desempenho

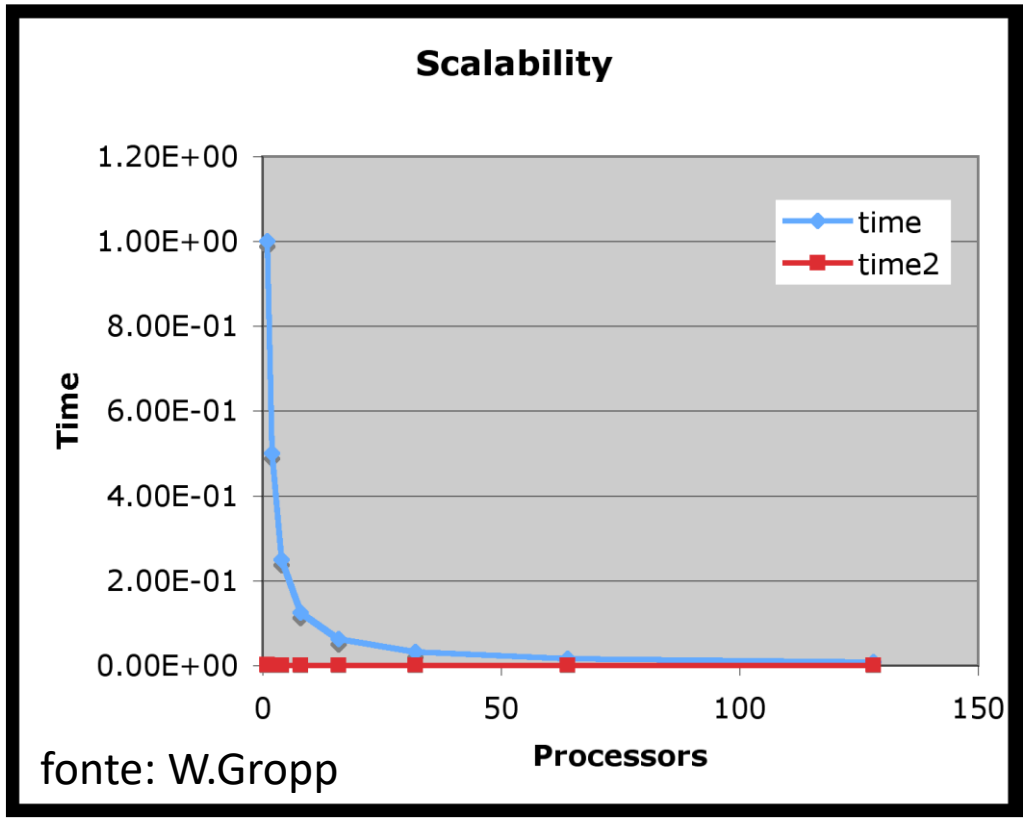
- Exemplo: dois algoritmos paralelos, (1) e (2)

  - Problema: Tempo de execução serial de (1) é muito maior que o de (2)!
    Logo, (2) é preferível!

  Conclusão: cuidado ao fazer comparações, mesmo olhando as mesmas métricas!

  "Fácil" ter boa escalabilidade:

  basta um cód.serial ineficiente!



Modelagem de Desempenho

- Utilidades: várias
  - Prever o desempenho de uma aplicação num sistema antes da execução em si
  - Comparação entre dois algoritmos num sistema
  - Comparação entre dois sistemas para um algoritmo
  - Descobrir partes críticas de uma aplicação
- Duas formas principais de modelagem
  - Simulação
  - Expressões analíticas

Modelagem de Desempenho

- Simulação:
  - Em geral, requer profundo conhecimento do sistema
  - Custo de execução pode ser alto, proibitivo em alguns casos
  - Útil para programadores de aplicações e projetistas de sistemas
  - Mais utilizada quando os sistemas não estão disponíveis
- Expressões Analíticas:
  - Podem ter nível de detalhamento variável
  - Podem evoluir à medida que se torne necessário
  - Mesmo quando imprecisas, são úteis para se estimar tendências

Modelagem de Desempenho

- Exemplo de Modelagem Analítica:
  - Multiplicação de matrizes: Algoritmo convencional

```
do i=1, n
    do j=1,n
            c(i,j) = 0
            do k=1,n
                    c(i,j) = c(i,j) + a(i,k) * b(k,j)
```

Para matrizes de tamanho N:

- N^3 iterações do loop *k*
- 2 operações aritméticas por iteração
- Total: $2.N^3$ operações, tempo = $2.N^3.f$ onde f=T(operação)

Modelagem de Desempenho

- Exemplo de Modelagem Analítica (cont.):
  - Tempo: $T = 2.N^3.f$
  - Assume que o tempo total é função de N e f apenas (isto é, do número de oper. aritméticas e da duração de cada operação)
  - Aumento de tamanho do problema: N→N'
    - Novo tempo será: $T' = 2.(N')^3.f$
    - Exemplo: se $N'=2.N \rightarrow T' = 2.(2.N)^3.f = 2.8.N^3.f = 8.T$
    - Em geral: $N'=K.N \rightarrow T'=K^3.T$
    - Porém, como os grupos de CAP-372 demonstraram, isto nem sempre ocorre na prática, principalmente para N e N' muito distantes, mesmo sob o mesmo compilador!
      - ➢ Modelo original não é preciso; por exemplo, ignora $T_{memória}$

Benchmarks

- Alternativa à modelagem para caracterizar desempenho
  - Programas que avaliam/medem algum aspecto do sistema
  - Aplicações completas ou trechos críticos
  - Códigos devem ser amplamente disponíveis
  - Regras claras para execução e divulgação de resultados
  - Uso:
    - comparação entre sistemas
    - “prova” de capacidade em processos de compra
      - deve representar bem a carga de trabalho esperada no sistema

Benchmarks (cont.)

- Exemplos atuais:
  - Linpack – solução de sistemas lineares densos
  - HPC Challenge (HPCC) - tentativa de extender o Linpack
  - STREAM – padrões de acesso à memória
  - NAS Benchmarks - extraídos de aplicações científicas reais
  - Graph500 – aplicação não-numérica: proc. de grafos
  - SPP (*Sustained Petaflop Performance*) – aplicações completas
- Outros tipos:
  - IMB: desempenho de funções de MPI
  - IOR: desempenho de operações de I/O

Benchmarks: Linpack

- Principais Características
  - Utilizado para çlassificação no Top500
  - Resolução de um sistema de *n* equações lineares através do método de Eliminação de Gauss
  - Tempo de execução da ordem de $2.N^3/3$
    - Valor de N deve ser informado, junto com os resultados
  - Uso de memória da ordem de N^2
  - Em geral, desempenho aumenta com N
  - Desempenho é medido pelo próprio código

Benchmarks: HPC Challenge

- Coleção de Benchmarks
  - HPLinpack – Linpack original
  - DGEMM – multiplicação de matrizes densas
  - STREAM – acesso à memória
  - PTRANS – transposição de matrizes
  - RandomAccess: acessos a posições de memória aleatórias
  - FFT: transformada em 1D
  - Comunicação: latência e largura de banda

Benchmarks: STREAM

- Velocidade de acesso à memória com 4 operações
  - COPY: x(i)=y(i)
  - SCALE: x(i)=a*y(i)
  - ADD: x(i)=y(i)+z(i)
  - TRIAD: x(i)=y(i)+a*z(i)
- Alvo principal:
  - Vetores com tamanho grande
    - Tal que os vetores não caibam nas caches
  - Resultados: medidos em *bytes/segundo*

Benchmarks: NAS

- NAS Benchmarks:
  - Baseados em aplicações utilizadas pela NASA
  - Códigos disponíveis em MPI ou MPI+OpenMP
  - Várias “classes” (tamanhos) de problema definidas
  - Diferentes tipos de códigos/operações:
    - Ordenação de inteiros
    - Gradiente conjugado
    - Multigrid
    - FFT 3D
    - Outros “solvers”

Benchmarks: Graph500

- Objetivos:
  - Avaliar sistemas na execução de processamento não-numérico
  - Complementar a lista Top500
- Benchmark:
  - Algoritmo de grafos - Breadth-First Search (BFS)
  - Implementações de referência em MPI e OpneMP
  - Métrica: GTEPS
    - Velocidade de percorrimento das arestas do grafo
  - Outros algoritmos são permitidos (desde que gerem resultados corretos)
  - Mais benchmarks estão em desenvolvimento

Benchmarks: SPP, etc

- SPP: Coleção de códigos reais
  - Objetivo: medir desempenho sustentado das aplicações
  - Códigos completos, com I/O, checkpoint, etc.
  - Índice final: média geométrica dos vários componentes
- Outros: Projeto CORAL (DOE-EUA)
  - Utilizado para a compra de sistemas para labs nacionais
  - De aplicações completas a microbenchmarks
  - Códigos em Fortran/C/C++/Python
  - Paradigmas utilizados: MPI/OpenMP/Pthreads

Obtenção dos Benchmarks

- Benchmarks tradicionais
  - Linpack: http://top500.org
  - HPC Challenge: http://icl.cs.utk.edu/hpcc
  - STREAM: https://www.cs.virginia.edu/stream/
  - NAS Benchmarks: http://www.nas.nasa.gov/publications/npb.html
  - Graph500: http://graph500.org
  - SPP (*Sustained Petaflop Performance*) – em construção
  - CORAL - https://asc.llnl.gov/CORAL-benchmarks/
- Outros tipos
  - IMB (Intel) - https://software.intel.com/en-us/articles/intel-mpi-benchmarks/
  - IOR: http://www.nersc.gov/users/computational-systems/cori/nersc-8-procurement/trinity-nersc-8-rfp/nersc-8-trinity-benchmarks/ior/