Divisible load theory

Loris Marchal, material from Frédéric Vivien

12 février 2009
Overview

Summary of the first lecture

Adaptation to tree-shaped platforms

Multi-round algorithms

With bounded memory

Case study: article from Veeravalli and Robertazzi

Conclusion
Overview

Summary of the first lecture

Adaptation to tree-shaped platforms

Multi-round algorithms

With bounded memory

Case study: article from Veeravali and Robertazzi

Conclusion
Summary of the first lecture

- Key assumption: work is *divisible* in rational quantities
- Renders many problems tractable
- Underlying idea: since the number of tasks is large, rounding to integer values after computing optimal schedule is negligible

- A schedule is described by:
  - the set of participating processors,
  - the order of the sending operations,
  - the quantity of work sent to each processor

- Basic problem: star network, linear costs, one-round,
  - All workers participate
  - Send work to workers with largest bandwidth first
  - All workers terminate at the same time: we are able to compute the amount of work done by each worker

Many possible generalizations.
Overview

Summary of the first lecture

Adaptation to tree-shaped platforms

Multi-round algorithms

With bounded memory

Case study: article from Veeravali and Robertazzi

Conclusion
Adaptation to tree-shaped platforms

- Each single level tree can be replaced by a single node, with total computing capacity $W$, with $w = \sum \alpha_i$, where $\alpha$ is the solution of the previous linear program.

- Constructive solution for the tree:
  1. Traverse the tree from bottom to top, replacing each single-level node by an equivalent processor.
  2. Solve the star problem obtained.
  3. Traverse the tree from top to bottom, undo each transformation, order the children, and distribute the load.

- Global solution: order the children by non-decreasing bandwidth, and then write the complete linear program.
Overview

Summary of the first lecture

Adaptation to tree-shaped platforms

Multi-round algorithms

With bounded memory

Case study : article from Veeravali and Robertazzi

Conclusion
One round vs. multi-round

**One round**

\[ \sim \text{long idle-times} \]

**Multi-round**

Efficient when \( W_{\text{total}} \) large

*Intuition*: Start with small rounds, then increase chunk sizes.

Problem with current model: leads to an absurd solution with infinite number of infinitely small messages.

- Either change the model in order to allow simultaneous communication and computation on the same data
- Or add latency to the communication the model
Notations

- A set $P_1, \ldots, P_p$ of processors
- $P_1$ is the master processor: initially, it holds all the data.
- The overall amount of work: $W_{\text{total}}$.
- Processor $P_i$ receives an amount of work $\alpha_i W_{\text{total}}$ with $\sum_i n_i = W_{\text{total}}$ with $\alpha_i W_{\text{total}} \in \mathbb{Q}$ and $\sum_i \alpha_i = 1$.
  
  Length of a unit-size work on processor $P_i$: $w_i$.
  Computation time on $P_i$: $n_i w_i$.

- **Time needed to send a message of size** $\alpha_i$ **from** $P_1$ **to** $P_i$: $L_i + c_i \times \alpha_i$.
  One-port model: $P_1$ sends and receives a single message at a time.
Complexity

Definition (One round, $\forall i, c_i = 0$)

Given $W_{total}$, $p$ workers, $(P_i)_{1 \leq i \leq p}$, $(L_i)_{1 \leq i \leq p}$, and a rational number $T \geq 0$, and assuming that bandwidths are infinite, is it possible to compute all $W_{total}$ load units within $T$ time units?

Theorem

The problem with one-round and infinite bandwidths is NP-complete.

What is the complexity of the general problem with finite bandwidths and several rounds?

The general problem is NP-hard, but does not appear to be in NP (no polynomial bound on the number of activations).
Fixed activation sequence

Hypotheses

1. Number of activations: $N_{\text{act}}$;
2. Whether $P_i$ is the processor used during activation $j$: $\chi_{i}^{(j)}$

Minimize $T$, under the constraints

\[
\begin{align*}
&\sum_{j=1}^{N_{\text{act}}} \sum_{i=1}^{p} \chi_{i}^{(j)} \alpha_{i}^{(j)} = W_{\text{total}} \\
&\forall k \leq N_{\text{act}}, \forall l : \left( \sum_{j=1}^{k} \sum_{i=1}^{p} \chi_{i}^{(j)} (L_i + \alpha_{i}^{(j)} c_i) \right) + \sum_{j=k}^{N_{\text{act}}} \chi_{l}^{(j)} \alpha_{l}^{(j)} w_l \leq T \\
&\forall i, j : \alpha_{i}^{(j)} \geq 0
\end{align*}
\]

(1)

Can be solved in polynomial time.
Fixed number of activations

Minimize $T$, under the constraints

\[
\sum_{j=1}^{N_{\text{act}}} \sum_{i=1}^{p} \chi_i^{(j)} \alpha_i^{(j)} = W_{\text{total}}
\]

\[
\forall k \leq N_{\text{act}}, \forall l : \left( \sum_{j=1}^{k} \sum_{i=1}^{p} \chi_i^{(j)} (L_i + \alpha_i^{(j)} c_i) \right) + \sum_{j=k}^{N_{\text{act}}} \chi_l^{(j)} \alpha_l^{(j)} w_l \leq T
\]

\[
\forall k \leq N_{\text{act}} : \sum_{i=1}^{p} \chi_i^{(k)} \leq 1
\]

\[
\forall i, j : \chi_i^{(j)} \in \{0, 1\}
\]

\[
\forall i, j : \alpha_i^{(j)} \geq 0
\]

Exact but exponential
Can lead to branch-and-bound algorithms
Uniform multi-round

In a round: all workers have same computation time

Geometrical increase of rounds size

No idle time in communications:

\[
\alpha_i^{(j)} w_i = \sum_{k=1}^{p} (L_k + \alpha_k^{(j+1)} c_k).
\]

Heuristic processor selection: by decreasing bandwidths

No guarantee...
How to choose $T_p$? Which resources to select?
With no overlap (1/4)

Equations

- Divide total execution time $T$ into $k$ periods of duration $T_p$.
- $\mathcal{I} \subset \{1, \ldots, p\}$ participating processors.
- Bandwidth limitation:
  \[ \sum_{i \in \mathcal{I}} (L_i + \alpha_i c_i) \leq T_p. \]
- No overlap:
  \[ \forall i \in \mathcal{I}, \quad L_i + \alpha_i (c_i + w_i) \leq T_p. \]
Normalization

- $\beta_i$: average number of tasks processed by $P_i$ during one time unit.

\[
\text{MAXIMIZE } \sum_{i=1}^{p} \beta_i \\
\forall i \in I, \quad \beta_i(c_i + w_i) \leq 1 - \frac{L_i}{T_p} \\
\sum_{i\in I} \beta_i c_i \leq 1 - \frac{\sum_{i\in I} L_i}{T_p}
\]

Relaxed version

\[
\text{MAXIMIZE } \sum_{i=1}^{p} x_i \\
\forall 1 \leq i \leq p, \quad x_i(c_i + w_i) \leq 1 - \frac{\sum_{i=1}^{p} L_i}{T_p}
\]

\[
\sum_{i=1}^{p} x_i c_i \leq 1 - \frac{\sum_{i=1}^{p} L_i}{T_p}
\]
Normalization

- $\beta_i$ average number of tasks processed by $P_i$ during one time unit.

$$\text{Maximize } \sum_{i=1}^{p} \beta_i$$

- Linear program:

$$\begin{align*}
\forall i \in I, \quad \beta_i(c_i + w_i) &\leq 1 - \frac{L_i}{T_p} \\
\sum_{i \in I} \beta_i c_i &\leq 1 - \frac{\sum_{i \in I} L_i}{T_p}
\end{align*}$$

Relaxed version

$$\text{Maximize } \sum_{i=1}^{p} x_i$$

$$\begin{align*}
\forall 1 \leq i \leq p, \quad x_i(c_i + w_i) &\leq 1 - \frac{L_i}{T_p} \\
\sum_{i=1}^{p} x_i c_i &\leq 1 - \frac{\sum_{i=1}^{p} L_i}{T_p}
\end{align*}$$
With no overlap (2/4)

Normalization

- $\beta_i$ average number of tasks processed by $P_i$ during one time unit.

$$\text{MAXIMIZE } \sum_{i=1}^{p} \beta_i$$

- Linear program:

$$\forall i \in I, \quad \beta_i(c_i + w_i) \leq 1 - \frac{L_i}{T_p}$$

$$\sum_{i \in I} \beta_i c_i \leq 1 - \frac{\sum_{i \in I} L_i}{T_p}$$

Relaxed version

$$\text{MAXIMIZE } \sum_{i=1}^{p} x_i$$

$$\forall 1 \leq i \leq p, \quad x_i(c_i + w_i) \leq 1 - \frac{\sum_{i=1}^{p} L_i}{T_p}$$

$$\sum_{i=1}^{p} x_i c_i \leq 1 - \frac{\sum_{i=1}^{p} L_i}{T_p}$$
Bandwidth-centric solution

- Sort: \( c_1 \leq c_2 \leq \ldots \leq c_p \).
- Let \( q \) be the largest index so that \( \sum_{i=1}^{q} \frac{c_i}{c_i + w_i} \leq 1 \).
- If \( q < p \), \( \epsilon = 1 - \sum_{i=1}^{q} \frac{c_i}{c_i + w_i} \).
- Optimal solution to relaxed program:

\[
\forall 1 \leq i \leq q, \quad x_i = \frac{1 - \sum_{i=1}^{p} \frac{L_i}{T_p}}{c_i + w_i}
\]

and (if \( q < p \)):

\[
x_{q+1} = \left(1 - \frac{\sum_{i=1}^{p} L_i}{T_p}\right) \left(\frac{\epsilon}{c_{q+1}}\right),
\]

and \( x_{q+2} = x_{q+3} = \ldots = x_p = 0 \).
Asymptotic optimality

- Let $T_p = \sqrt{T_{\text{max}}^*}$ and $\alpha_i = x_i T_p$ for all $i$.
- Then $T \leq T_{\text{max}}^* + O(\sqrt{T_{\text{max}}^*})$.
- Closed-form expressions for resource selection and task assignment provided by the algorithm.
With overlap

Key points

◀ Still sort resources according to the $c_i$.

◀ Greedily select resources until the sum of the ratios $\frac{c_i}{w_i}$
  \(\text{(instead of } \frac{c_i}{c_i+w_i})\) exceeds 1.
Overview

Summary of the first lecture

Adaptation to tree-shaped platforms

Multi-round algorithms

With bounded memory

Case study : article from Veeravali and Robertazzi

Conclusion
Divisible load scheduling with bounded memory

- Assume the memory is bounded on each worker
- Problem is NP-complete with affine costs (reduction from 3-partition)
Overview

Summary of the first lecture

Adaptation to tree-shaped platforms

Multi-round algorithms

With bounded memory

Case study: article from Veeravalli and Robertazzi

Conclusion
Overview

Summary of the first lecture

Adaptation to tree-shaped platforms

Multi-round algorithms

With bounded memory

Case study: article from Veeravali and Robertazzi

Conclusion
Que retenir de tout ça ?

▶ Idée de base simple : une solution approchée est amplement suffisante.

▶ Les temps de communication jouent un plus grand rôle que les vitesses de calcul.