Scheduling divisible workloads on heterogeneous platforms under Bounded Multi-Port model

Olivier Beaumont, Nicolas Bonichon, Lionel Eyraud-Dubois

LAboratoire Bordelais de Recherche en Informatique
CEPAGE INRIA Team-Project

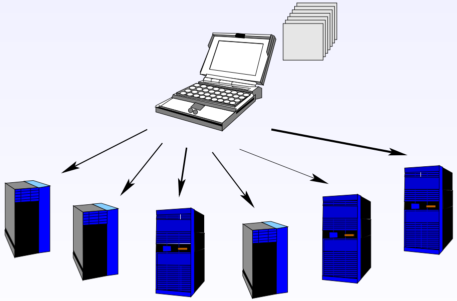
Scheduling in Aussois
May 2008

Outline

Outline

Introduction

- One master, holding a large number of identical tasks
- Some workers
- Heterogeneity in computing speed and bandwidth
- Distribute work to workers



Assumption: divisible load

- Important relaxation of the problem
- Master holding N tasks
- Worker P_i will get a fraction $\alpha_i \times N$ of these tasks
- α_i is **rational**, tasks are divisible
- $\Rightarrow$ possible to derive analytical solutions (tractability)
- In practice, reasonable assumption with a large number of tasks
- Worker P_i can start processing tasks only once it has received the whole data (1-round)

Background on Divisible Load Scheduling: 1-port

- Linear cost model: X units of work:
  - sent to P_i in $X \times c_i$ time units
  - computed by P_i in $X \times w_i$ time units

Background on Divisible Load Scheduling: 1-port

- Linear cost model: X units of work:
  - sent to P_i in $X \times c_i$ time units
  - computed by P_i in $X \times w_i$ time units

Results:

- Bus network $\Rightarrow$ all processors work and finish at the same time order does not matter closed-formula for the makespan (Bataineh, Hsiung & Robertazzi, 1994)
- Heterogeneous star network:
  all processors work and finish at the same time
  order matters:
  - largest bandwidth first (whatever the computing power)

Background on Divisible Load Scheduling: 1-port

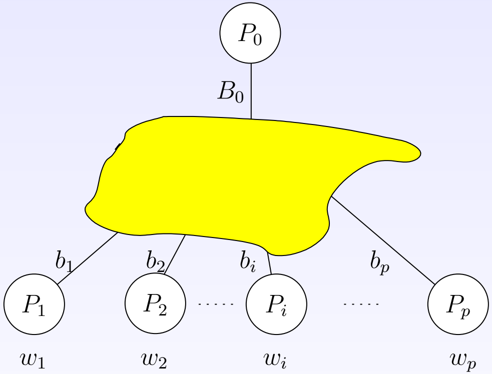
- With an affine cost model: X units of work:
  - sent to P_i in $C_i + X \times c_i$ time units
  - computed by P_i in $W_i + X \times w_i$ time units

$\Rightarrow$ not all processors participate to the computation
selecting the resources is hard:

- computing the optimal schedule on a star with affine cost model is NP-hard (Casanova, Drodowski, Legrand & Yang, 2005)

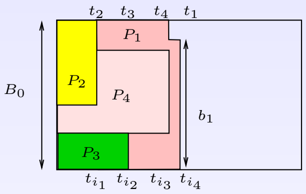
Outline

Platform



- B_0: output bandwidth of the master processor.
- b_i: input bandwidth of the slave P_i.
- w_i: time needed by P_i to compte a unit-task.
- Assumption: No interaction between communications.

Bounded Multi-Port Model



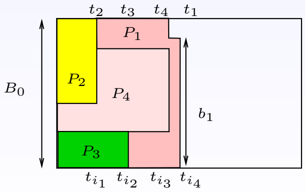
- Notations:
  - $b'_i(t)$: actual bandwidth used at time t by the communication between P_0 and P_i.
  - t_i: the time when processor P_i stops communicating.
  - q_i: the fractional number of tasks processed by P_i: $q_i = \int_0^1 \sum_i b'_i(t)dt$.
- Constraints:
  - input bandwidth: $\forall t, b'_i(t) \leqslant b_i$.
  - output bandwidth: $\forall t, \sum_i b'_i(t) \leqslant B_0$.
  - processing time: $t_i + w_i.q_i \leqslant 1$
- Goal: Maximizing $\sum_i q_i$

Normal Form

Definition

A schedule is said to be in *normal form* if

1. all processors are involved in the processing of tasks,
2. all slaves start processing tasks immediately after the end of the communication with the master (at time t_i) and stop processing at time 1,
3. during each time slot $]t_{i_k}, t_{i_{k+1}}]$ the bandwidth used by any processor is constant.

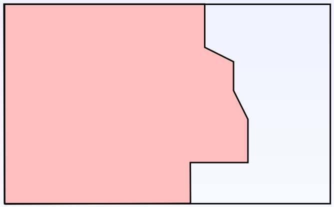


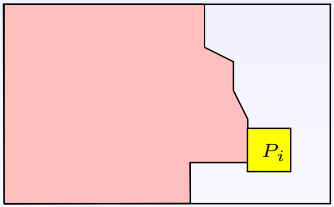
Lemma 1

Lemma

In an optimal schedule, all processors take part in the computations.

Proof:


Original schedule



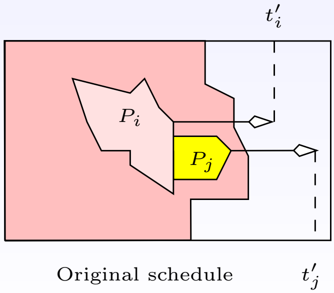
New schedule

Lemma 2

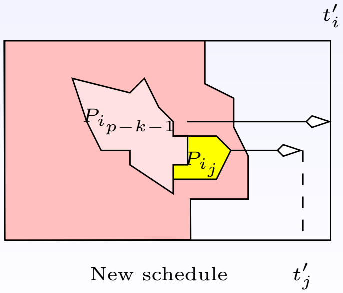
Lemma

In an optimal schedule, slaves start processing tasks immediately after the end of the communication with the master and stop processing at time 1, i.e. there is no idle time between the end of the communication and the deadline.

Proof: By induction on the first slave that stop processing before $t = 1$.



Original schedule



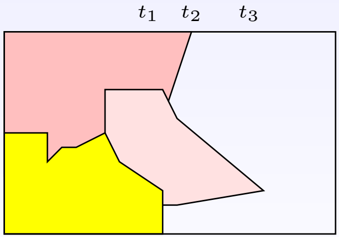
New schedule

Lemma 3

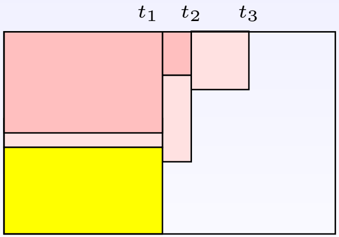
Lemma

There exists an optimal schedule such that during each time slot $]t_{i_k}, t_{i_{k+1}}]$ *the bandwidth used by any processor is constant.*

Proof:



Original schedule



New schedule

Normal Form

Theorem

We can restrict the search of optimal solutions within the valid schedules in normal form.

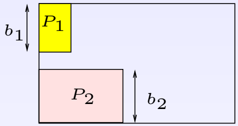
Proof.

The proof of the theorem comes directly from Lemmas 1, 2 and 3. □

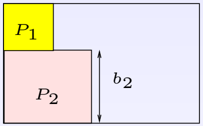
Outline

Different cases with 2 slaves

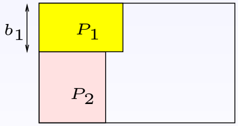
$b_2 < B_0 \frac{w_1}{w_1+w_2}$ and $b_1 < B_0 \frac{w_2}{w_1+w_2}$



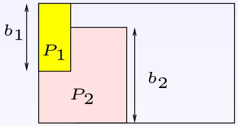
$b_2 < B_0 \frac{w_1}{w_1+w_2}$ and $b_1 \geqslant B_0 \frac{w_2}{w_1+w_2}$

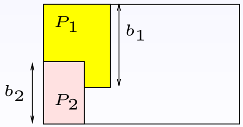


$b_2 \geqslant B_0 \frac{w_1}{w_1+w_2}$ and $b_1 < B_0 \frac{w_2}{w_1+w_2}$

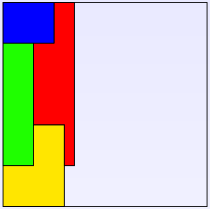


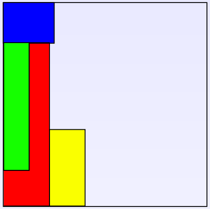
$b_2 \geqslant B_0 \frac{w_1}{w_1+w_2}$ and $b_1 \geqslant B_0 \frac{w_2}{w_1+w_2}$

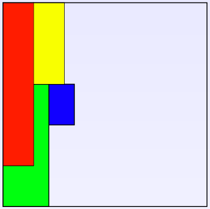


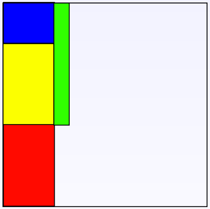


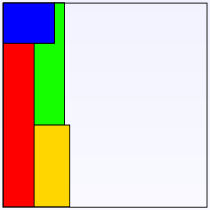
Extension of 1-port solutions ?

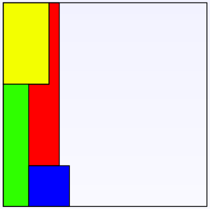

1324 (decreasing w): 2.067


1342 (decreasing w): 2.03


4321 (decreasing b): 2.042


1243 (increasing $b * w$): 2.03


1423 (best solution): 2.078


2314 (best solution): 2.078

$B_0 = 5$

🟦 $1 : b = 1, w = 2$
🟨 $2 : b = 2, w = 1$
🟩 $3 : b = 3, w = 1.5$
🟥 $4 : b = 4, w = 1$

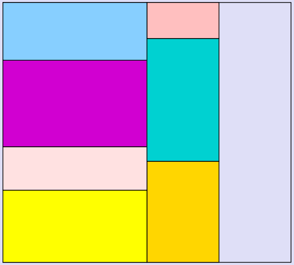
Outline

Theorem

BOUNDEDMPDIVISIBLE *is NP-complete.*

Proof.

Reduction from 2-PARTITION. We build an instance of BOUNDEDMPDIVISIBLE with $w_i = 1/b_i$ and $\sum_i b_i = 2B_0$.

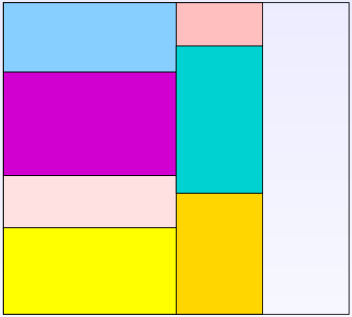


$Q = 3B_0/4$

□

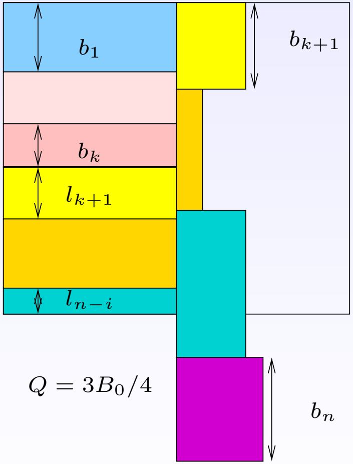
NP-Completeness: proof

(a) Optimal solution



$Q = 3B_0/4$

(b) Hypothetic solution



$Q = 3B_0/4$

Outline

Priority-Based Schedules

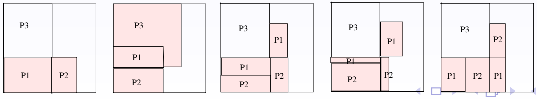
Priority-based Schedules

- a priority is assigned to each processor
- P_i gets at any time the maximal bandwidth
  - given its capability
  - given the bandwidth left by $P_1 \ldots P_{i_1}$

Theorem

In the case of 2 processors, there always exists an optimal priority-based schedule

- The theorem with more than 2 processors is open!
- There are optimal non-priority-based scheduling $(4, \frac{1}{4}), (4, \frac{1}{4}), (8, \frac{1}{8}), B_0 = 12$



Stability Issues

- Finding the optimal ordering under 1-port model is easy
- but a bad guess may lead to arbitrarily bad results
  - Consider $P_1 = (1, 1)$ and $P_2 = (\frac{1}{\varepsilon}, 1)$
  - if you schedule P_1 first: $\frac{1+\varepsilon}{2}$ tasks
  - if you schedule P_2 first: $\frac{3\varepsilon}{2}$ tasks only!
- Under bounded multi-port model,
  - finding the optimal schedule is hard (NP-Complete)
  - but the worst ratio is $\frac{8}{9}$ (purely analytic result) between priority-based schedules (in the case of 2 processors)!

what may be useful if bandwidth estimation are unreliable!

- In the case of BMP, the "almost only" important thing is to correctly estimate processing time to stop the communication at the right time!

Open Questions

Theoretical

- Is there always a priority-based optimal schedule? (Ok for 2 procs)
- What is the worst possible ratio among priority schedule? ($\frac{8}{9}$ for 2 processors)
- Is the problem strongly NP Complete
    - the reduction to 3-partition does not work...

Practical

- one-round is not efficient, but multi-round is stupid without latencies
    - how to add latencies in a reasonable way (not just to prove NP-Completeness)?
    - and how to prove approximation ratio in this case?
- How to implement prescribed sharing of bandwidth between concurrent communications? (and how to do it in user mode?)

Outline

Conclusion

- A more realistic model for Divisible Load Scheduling
- Exhaustive study with two processors $\Rightarrow$: no intuition :-(
- NP-Completeness result without latencies
- Still many open questions!