INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Parallel Data Redistribution Over a Backbone

Johanne Cohen — Emmanuel Jeannot — Nicolas Padoy

N° 4725

Février 2003

THÈME 1

Rapport de recherche

N 0249-6399

Parallel Data Redistribution Over a Backbone

Johanne Cohen* , Emmanuel Jeannot† , Nicolas Padoy‡

Thème 1 — Réseaux et systèmes
Projet AlGorille

Rapport de recherche n° 4725 — Février 2003 — 14 pages

Abstract: In this report we study the general problem of parallel data redistribution over a network. Given a set of communications between two parallel machines interconnected by a backbone, we wish to minimize the total time required for the completion of all communications assuming that communications can be preempted and that preemption comes with an extra cost. Our problem, called *k-Preemptive bipartite scheduling (KPBS)* is proven to be NP-Complete. Moreover we prove that approximating KPBS problem within a ratio number smaller that $\frac{4}{3}$ is impossible unless $P = NP$. In spite of this negative result, we study a lower bound of this problem, and we propose an approximation algorithm with ratio 2 and fast heuristics.

Key-words: Data redistribution, inaproximability, approximation algorithm grid computing

* CNRS-INRIA Lorraine
† Université Henri Poincaré, LORIA-INRIA Lorraine
‡ École Normale Supérieure de Lyon

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,

Redistribution parallèle de données à travers un backbone

Résumé : Dans ce rapport nous étudions le problème général de la redistribution de données à travers un réseau. Etant donné un ensemble de communications et deux machines parallèles interconnectées par un backbone. L'objectif est de minimiser le temps de terminaison de toutes les communications en supposants que celle-ci peuvent-être préemptées et que la préemption a un coût. Nous provons que notre problème, appelé *k-Preemptive bipartite scheduling (KPBS)* est NP-Complet. De plus nous provons qu'il est impossible d'approximer KPBS d'un facteur plus petit que 4/3 à moins que P=NP. Malgré ce résultat négatif, nous étudions la borne inferieure de ce problème et nousproposons un algorithme 2-approché ainsi que des heuristiques rapides.

Mots-clés : Redistribution de données, inaproximation, algorithme d'approximation, calcul sur la grille

1 Introduction

With the emergence of grid computing many scientific applications use code coupling technologies to achieve their computations where parts of the code are distributed among parallel resources interconnected by a network. Code coupling requires data to be redistributed from one parallel machine to another. For instance the NxM ORNL project [11] has for objective to specify a parallel data redistribution interface and CUMULVS [9] (which uses MxN) supports interactive and remote visualization of images generated by a parallel computer. The parallel data redistribution is realized on a network, called a backbone, which interconnects the two parallel machines. If the parallel redistribution pattern involves a lot of data transfers, therefore the backbone can become a bottleneck. Thus, in order to optimize the parallel data redistribution time and to avoid the overloading of the backbone it is required to schedule each data transfer.

Data redistribution has mainly been studied in the context of high performance parallel computing [5, 1, 6]. In this paper we study a generalization of the parallel data redistribution. Indeed, contrary to some previous works that only deal with block-cyclic redistribution [6, 3], here, no assumption is made on the redistribution pattern. Moreover, contrary to other works which assume that there is no bottleneck [5, 1], we suppose that the ratio between the throughput of the backbone and the throughput of each of the n nodes of the parallel machines is k. Hence, no more than k communications can take place at the same time. We study the problem for all values of k. We focus on the case $k < n$ (the backbone is a bottleneck) whereas the case $k \geq n$ has been tackled in [1, 5].

The contribution of this paper is the following. We prove that the problem of scheduling any parallel data redistribution pattern is NP-Complete for any value of k ($< n$) and that approximating our problem (called KPBS) within a ratio number smaller that $\frac{4}{3}$ is impossible unless $P = NP$. We exhibit a lower bound for the number of steps of the redistribution as well as a lower bound for the sum of the duration of each step and prove that both lower bounds are tight. Next, we propose two algorithms: a pseudo-polynomial approximation algorithm with ratio 2, and polynomial approximation algorithm with ratio 2. Finally, we study simple and fast heuristics which achieve a good average performance.

The remaining of this paper is organized as follow. In section 2 we provide definitions as well as the modelization of the problem. Previous works are described in Section 3. In Section 4, we focus on the complexity of the problem and prove that it is NP-Complete for any k. In section 5, we show that approximating KPBS problem within a ratio number smaller that $\frac{4}{3}$ is impossible unless $P = NP$. In Section 6 we study a lower bound. We propose our approximation algorithm in Section 7. Finally, in Section 8 two simple heuristics are studied.

2 Definitions

2.1 Network constraints

We present the constraints relative to the network topology. We consider two clusters of workstations $\mathcal{G}_1$ and $\mathcal{G}_2$ connected together by a backbone of throughput D. We denote the minimum of the number of nodes over theses two clusters by n. All the nodes of the first cluster have a throughput d' and the nodes of the second have a throughput d''.

For instance, let us consider the following architecture. A cluster $\mathcal{G}_1$ of 50 PC's with 100 Mbps network cards is connected to a cluster $\mathcal{G}_2$ of 20 PC's with 10 Mbps network cards, using a 1 Gbps backbone. Thus $n = 20$, $d' = 100$, $d'' = 10$ and $D = 1000$.

We assume that the communication pattern of the redistribution is given. This pattern is model ed by a *traffic matrix* $Q = (q_{i,j})_{1 \leq i,j \leq n}$, where $q_{i,j}$ represents the amount of information that must be exchanged between the node i of cluster $\mathcal{G}_1$ and the node j of cluster $\mathcal{G}_2$.

For a given traffic pattern and for this particular architecture (see fig.1) our goal is to minimize the total transmission time.

We now explain the constraints relative to the communications. A transmitter (resp. receiver) cannot transmit (resp. receive) more than one message at a time. However we allow the transmission of several messages between different transmitters and receivers simultaneously as long as the backbone is not saturated. A parallel transmission *step* is a communication phase in which there can be simultaneous transmissions

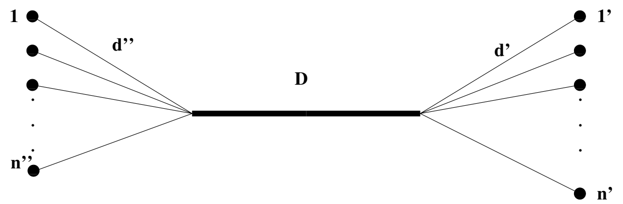

Figure 1: Initial problem

between several transmitters and receivers. A common approach to minimize the overall transmission time is to allow preemption, i.e the possibility to interrupt the transmission of a message and complete it later. In practice, this involves a non negligible cost, called *set-up delay* and denoted here by β, which is the time necessary to start a new *step*. we set $d = \min(d', d'')$ as the maximum number of machines involved in a communication step at the same time, on both side.

2.2 Formulation of the problem

Before giving the formulation of the problem, we will discuss the relationship between parameters D and d.

- If $d \geq D$, the backbone is saturated by one communication, which is not an interesting case.
- If $n \times d \leq D$, the backbone is not a bottleneck. This case has already been studied in [1, 5].
- If $1 < \frac{D}{d} < n$, then the number of communications per step is bounded by $\lfloor \frac{D}{d} \rfloor$.

Thus, we only consider the case where $1 < \frac{D}{d} \leq n$. We can simplify the notation. Each step can be composed of at most $k = \lfloor \frac{D}{d} \rfloor$ communications. Moreover, the traffic matrix Q, can be replaced by the matrix $A = (a_{i,j}) = (\frac{q_{i,j}}{d})_{1 \leq i \leq n, 1 \leq j \leq n}$ which represents the communication times between the nodes of the two clusters.

This problem can be represented by a bipartite graph $G = (V_1, V_2, E)$ and a positive edge-weight function $w : E \to \mathbb{Q}$. Each node of cluster $\mathcal{G}_1$ (resp. $\mathcal{G}_2$) is represented by a node of set V_1 (resp. V_2). The weight of an edge is the time of the communication that must be achieved between the two nodes.

A communication step is a weighted matching of k edges since a node takes part in at most one communication at a time and a step has at most k communications. The weights refer to preemption. We denote the matching corresponding to a communication step by a *valid weighted matching* (for the remainder, a valid weighted matching contains at most k edges).

We call this problem *k-Preemptive bipartite scheduling (KPBS)*, formally defined as follows:

Given a weighted bipartite graph $G = (V_1, V_2, E, w)$ where $w : E \to \mathbb{Q}$, an integer k and a rational number β, find a collection $\{(M_1, W_1), (M_2, W_2), \ldots, (M_s, W_s)\}$ of valid weighted matching such that

1. for any $1 \leq i \leq s$, matching M_i has at most k edges ($|M_i| \leq k$) and its weight is equal to the rational number $W_i = \max_{e \in M_i} w_i(e)$.
2. We define w_i the edge weight function of each matching M_i as follows. For any $e \in E$, $\sum_{i=1}^{s} w_i(e) \geq w(e)$ where $w_i(e) < W_i$ if $e \in M_i$, $w_i(e) = 0$ otherwise.
3. $\sum_{i=1}^{s} W_i) + \beta \times s$ is minimized.

In the remainder of this paper, note that for any solution S of $KPBS$, if the cost of S is $\alpha + t\beta$, the number of steps of S is t and the useful transmission time of S equals to α. See Figure 2 for an example.

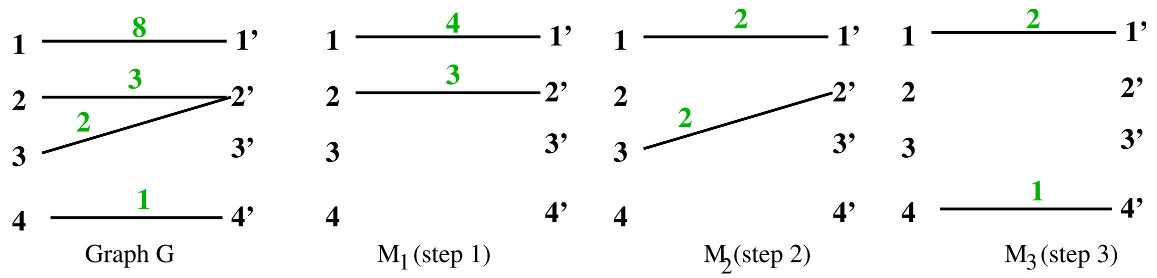

Figure 2: An example for KPPS problem ($k = 2$). The cost of the solution is $8 + 3\beta$

3 Previous Work

Up to our knowledge, there is no work on the KPBS problem. However, a particular case of problem KPBS (the case where $k = n$) has been studied in the literature [1, 5] and it is known as the *preemptive bipartite scheduling (PBS)*. PBS was proven to be NP-complete for a restricted class of instance [7, 10]. Moreover, approximating the KPBS problem within a ratio number smaller than $\frac{7}{6}$ is impossible unless $P = NP$ [5]. In spite of this negative result, there are some approximation algorithms for this problem. In [5], two different polynomial time 2-approximation algorithms for PBS have been proposed and in [1], the authors have given a small improvement of this result.

4 Problem Complexity

This problem has already been proven NP-complete for the particular case where $k = n$ [7, 10]. We prove that it remains NP-complete even though k is a fixed integer greater than two. The decision problem KPBS is defined as follow:

Instance: A weighted bipartite graph $G = (V_1, V_2, E, w)$ where $w : E \to \mathbb{Q}$, an integer k a rational number β, a rational number B.
Question: Is there a collection $\{(M_1, W_1), (M_2, W_2), \ldots, (M_s, W_s)\}$ of valid weighted matchings such that $G = \sum_{i=1}^{t} M_i$, and $\sum_{i=1}^{t} w(M_i) + t \times \beta \le B$.

Theorem 1 *Let $k \ge 2$ and $\beta > 0$ be fixed integers. KPBS is NP-complete.*

Proof of theorem 1: It is easy to see that KPBS belongs to NP. We provide a transformation from the *Partition* problem [8]:

Instance: A finite set $U = \{u_1, u_2, \ldots, u_m\}$ and a size $s(u) \in Z^+$ for each $u \in U$.
Question: Is there a subset $U_1 \subseteq U$ such that $\sum_{u \in U_1} s(u) = \sum_{u \in U - U_1} s(u)$?

Let $s(U) = \sum_{i=1}^{m} s(u_i)$. We transform partition to KPBS: Let $U = \{u_1, u_2, \ldots, u_m\}$ be a finite set and a size $s(u) \in Z^+$ for each $u \in U$ in an arbitrary instance of *Partition*. We must construct an instance of KPBS. The integer β has an arbitrary value. We set $B = s(U) + \beta \times m$ and $k = 2$. We consider the following weighted bipartite graph $G = (V_1, V_2, E, w)$:

- $V_1 = \{v_1, v_2, \ldots, v_m, x, y\}$ and $V_2 = \{g, h\}$
- $E = \{(x, h), (y, h), (v_i, g),$ for $1 \le i \le m\}$
- $w(x, h) = s(U)/2$ and $w(x, g) = s(U)/2$, $w(v_i, g) = s(u_i)$ for $1 \le i \le m$

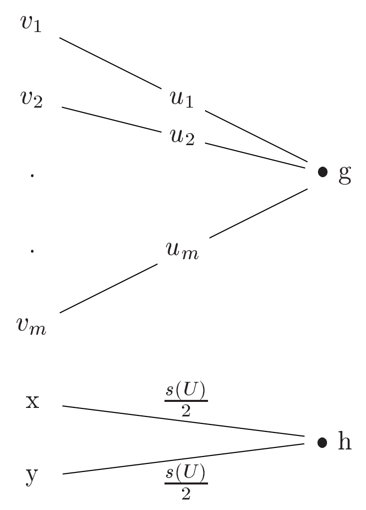

Figure 3: An example of graph G from a instance (U, s) of Partition Problem.

Figure 3 gives an example of this transformation. G can clearly be constructed in polynomial time. We claim that the instance of KPBS admits a solution if and only if the instance of Partition has a solution:

($\Rightarrow$) Let U_1 be a solution of the *Partition* instance. Then the valid matchings $\{(v_i, g)(x, h)\}_i$ having weights $s(u_i)$ with i such that $u_i \in U_1$, along with the valid matchings $\{(v_j, g)(x, h)\}_j$ having weights $s(u_j)$ with j such that $u_j \in U - U_1$ are a solution of KPBS. Indeed the sum of these matchings is $s(U)(\sum_{u_i \in U_1} s(u_i) + \sum_{u_i \in U - U_1} s(u_i) = s(U))$. The cost of this solution is exactly B.

($\Leftarrow$) Conversely, we suppose that the instance of KPBS admits a solution. Then the useful transmission time is at least equal to $s(U)$ because of the edges incident to vertex h. There are at least m steps, because the edges coming from the v_i are incident. Since the cost is lower than B, both previous inequalities are equalities. Therefore no edge incident to a v_i can be split. The size of a matching is at most $2(= k)$ and because the cost equals B, each matching C of the solution contains an edge incident to a v_i such that $w(C) \leq s(u_i)$ for the same i. Thus the matchings of the solution can be written $(C_i)_{1 \leq i \leq m}$. Necessarily each C_i contains an edge adjacent to x or y, having the weight $s(u_i)$ (the same weight as the other edge of the matching), in order to exhaust the weights of x and y.

Let U_1 be the set of the u_i such that C_i contains an edge adjacent to x. Then, since $w(x, h) = \frac{s(U)}{2}$, $s(U_1) = \frac{s(U)}{2} = s(U - U_1)$.

Since for any k fixed, we have the same proof, theorem 1 is proven.

□

5 Inapproximability

In this section, we improve the result in [5]. We prove that one cannot approximate the problem KPBS within a factor smaller than 4/3 if $P \neq NP$. We modify the reduction in [5] given in [7, 10]

Restricted Timetable Design Problem

Instance: A set $T = \{t_1, \ldots, t_\ell\}$ of teachers, a set $C = \{c_1, \ldots, c_m\}$ of classes, an availability function $A : T \times \{1, 2, 3\} \to \{0, 1\}$, et a requirement function $R : T \times C \to \{0, 1\}$

Question: Does there exist an assignment function $f : T \times C \times \{1, 2, 3\} \to \{0, 1\}$ such that

1. $f(t_i, c_j, k) = 1$ implies $A(t_i, k) = 1$

2. $\forall c_j \in C$, $\forall k \in \{1,2,3\}$, $\sum_{i=1}^{s} f(t_i, c_j, k) \leq 1$.
3. $\forall t_i \in T$, $\forall k \in \{1,2,3\}$, $\sum_{j=1}^{m} f(t_i, c_j, k) \leq 1$.
4. $\forall c_j \in C$, $\forall t_i \in T$, $\sum_{k=1}^{3} f(t_i, c_j, k) = R(t_i, c_j)$.

Figure 4 is an example of an instance Restricted Timetable Design Problem given in [5].

$$\text{A} = \begin{array}{|c|c|c|}\hline 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline \end{array} \quad \text{and} \quad \text{R} = \begin{array}{|c|c|c|c|}\hline 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ \hline \end{array}$$

An assignment function f:
$f(t_1, c_1, 2) = 1$
$f(t_1, c_3, 3) = 1$
$f(t_2, c_2, 1) = 1$
$f(t_3, c_3, 1) = 1$
$f(t_3, c_4, 2) = 1$

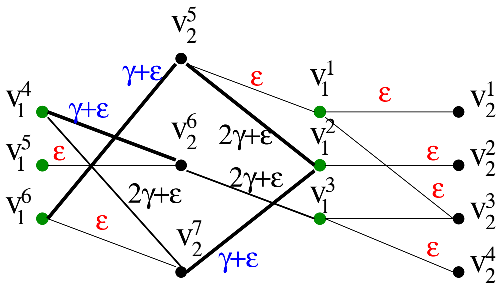

Figure 4: An example of graph G from an instance of Restricted Timetable Design Problem.

Let γ be an integer greater than 2. Given an arbitrary instance $I = \langle T, C, A, R \rangle$ of Restricted Timetable Design problem, we will define the corresponding instance $I' = \langle G = (V_1, V_2, E, w), k, \beta \rangle$ of KPBS as follows:

i. $k = \ell + m$. ϵ and β are any arbitrarily small rational number.

ii. $V_1 = \{v_1^1, \ldots, v_1^{2\ell}\}$, $V_2 = \{v_2^1, \ldots, v_2^{\ell+m}\}$ where $|V_1| = 2|T| = 2\ell$ and $|V_2| = |T| + |C| = \ell + m$

iii. $\forall i, j$, $1 \leq i \leq \ell$ and $1 \leq j \leq m$, $(v_1^i, v_2^j) \in E$ if and only if $R(t_i, c_j) = 1$. In this case, $w(v_1^i, v_2^j) = \epsilon$

iv. $\forall i, j$, $s+1 \leq i \leq 2\ell$ and $1 \leq j \leq m$, $(v_1^i, v_2^j) \notin E$

v. $\forall i, j$, $1 \leq i \leq \ell$ and $m+1 \leq j \leq m+\ell$, $\forall k \in \{1,2,3\}$, $(v_1^i, v_2^j) \in E$ if and only if $j = m + [(i+k-2) \bmod \ell + 1]$ and $A(t_i, k) = 0$. In this case, $w(v_1^i, v_2^j) = \gamma(k-1) + \epsilon$.

vi. $\forall i, j$, $\ell+1 \leq i \leq 2\ell$ and $m+1 \leq j \leq m+\ell$, $\forall k \in \{1,2,3\}$, $(v_1^i, v_2^j) \in E$ if and only if $j = m + [(i-\ell+k-2) \bmod s + 1]$ and $A(t_i, k) = 1$. In this case, $w(v_1^i, v_2^j) = \gamma(k-1) + \epsilon$.

Proposition 1 *Let $opt(I')$ be the optimal solution of instance I' of problem KBPS. If instance I of Restricted Timetable Problem admits a positif answer then, $opt(I') = 3\beta + 3\gamma + 3\epsilon$. Otherwise, $opt(I') \geq 3\beta + 4\gamma + 3\epsilon$.*

Proof of proposition 1 : Assume that f is an assignment function for I. For $h = 1,2,3$, we can define

$$\begin{aligned} M_h &= \{(v_1^i, v_2^j) : 1 \leq i \leq \ell, 1 \leq j \leq m, f(t_i, c_j, h) = 1\} \\ &\cup \{(v_1^i, v_2^j) : 1 \leq i \leq 2\ell, m+1 \leq j \leq m+\ell, w(v_1^i, v_2^j) = \gamma(h-1) + \epsilon\} \end{aligned}$$

By construction, $\{M_1, M_2, M_3\}$ is a solution of KPBS and its cost is equal to $3\beta + 3\gamma + 3\epsilon$.

Now, we will show by contradiction that, if I does not have as assignment function, then $opt(I') \geq 3\beta + 4\gamma + 3\epsilon$. So assume that I does not have an assignment function and $opt(I') < 3\beta + 4\gamma + 3\epsilon$. Since the minimum degree of G is 3, the number of steps is greater than 3. So since $opt(I') < 3\beta + 4\gamma + 3\epsilon$, the sum of the weights of all the matchings is less than $4\gamma + 3\epsilon$.

Since G has ℓ edges whose weight is $2\gamma + \epsilon$, at most one matching M contains edges whose weight is $2\gamma + \epsilon$. If it is not the case, then there are at least two matchings containing edges whose weight is $2\gamma + \epsilon$ and the sum of the weights of these matchings is $4\gamma + 2\epsilon$. All edges (v_1^i, v_2^j) in this matching M such that $1 \leq i \leq \ell$, $1 \leq j \leq m$ can construct partially an assignment function f: if $(v_1^i, v_2^j) \in M$ such that $1 \leq i \leq \ell$, $1 \leq j \leq m$, then $f(t_i, c_j, 2) = 1$. We can apply the same argument for the edges of weight $\gamma + \epsilon$ and of weight ϵ. Afterward, it is easy to check whether f is an assignment function satisfiyng properties 1, 2, 3 and 4.

□

Since, ϵ and β are any arbitrarily small rational number, from Proposition 1, using gap reduction technique (see [8, 14]), we can deduce that:

Theorem 2 *If $P \neq NP$, no polynomial time approximation algorithm for problem $KPBS$ with an approximation ratio smaller that $4/3$.*

6 Lower Bounds

Before giving a lower bound for the optimal solution, we give some graph notations. We define the weight $w(v)$ of a node v of G to be the sum of weights of all edges incident to vertex v. We denote the maximum of $w(v)$ over all vertices by $W(G)$. Let $P(G)$ be the sum of the weights of all edges of graph G. We denote the maximum degree of the bipartite graph G by $\Delta(G)$, its number of edges by $m(G)$ and its number of vertices by $n(G)$.

Proposition 2 *Let $G = (V_1, V_2, E, w)$ be a weighted bipartite graph. Let k and β be two integers. The cost of the optimal solution for the instance $\langle G, k, \beta\rangle$ of KPBS is at least $\eta(G) = \eta_d(G) + \beta \times \eta_s(G)$ where*

$$\eta_d(G) = max\left(W(G), \left\lceil \frac{P(G)}{k} \right\rceil\right) \quad and \quad \eta_s(G) = max\left(\Delta(G), \left\lceil \frac{m(G)}{k} \right\rceil\right)$$

Proof: $\eta_s(G)$ is a lower bound for the number of steps. The first term of the maximum accounts for the fact that two edges incident to the same node cannot appear in the same step and the second term for the fact that a step contains at most k edges. $\eta_d(G)$ is a lower bound for the useful transmission time and is obtained similarly. The total cost is therefore minimized by $\eta_d(G) + \beta \times \eta_s(G)$.

□

Next, we study the quality of these lower bounds. The remainder of this section is to prove that there are polynomial time algorithms to optimize the number of step (see Proposition 4) or the useful transmission time (see Proposition 3).

Proposition 3 *Let G be a weighted bipartite hypergraph. Then G can be decomposed such that the total transmission time is $\eta_d(G)$.*

Proposition 4 *Let G be a weighted bipartite hypergraph. Then G can be decomposed in $\eta_s(G)$ valid weighted matchings.*

Propositions 4 and 3 are equivalent. Indeed by setting all the weights to 1, Proposition 3 minimizes the number of steps because, in that case it is equal to the total transmission time. On the contrary, by splitting all the edges into edges of weight 1, Proposition 4 gives a solution which optimize the total transmission time. We present a similar polynomial-time algorithm for Proposition 4 which will be used later.
Remark: The previous propositions can be seen as a consequence (see [13]) of a coloration theorem (given in [2]).

We assume $k|m(G)$, adding some edges if necessary (since the lower bound η_s doesn't change). The proof of the following lemma is inspired from a similar matrix result in [4].

Lemma 1 *Let G be a bipartite hypergraph. Let $m(G)$ be the number of edges of G. If $\Delta(G) \leq \frac{m(G)}{k}$, then G has a matching of size k such that the possible vertices of degree $\frac{m(G)}{k}$ are saturated.*

Proof: Let $G = (X, Y, E, w)$. We add $n-k$ vertices to X and $n-k$ vertices to Y, which gives a graph $G^{'} = (X^{'}, Y^{'}, E^{'}, w^{'})$. Then we add edges to G' so that G' becomes a $\frac{m(G)}{k}$-regular hypergraph. Since $m(G)$ edges are coming out of X, $n \times \frac{m(G)}{k} - m(G)$ edges must be added to it, and it can be done because it is equal to $(n-k) \times \frac{m(G)}{k}$, which is the number of edges required for $Y^{'} \setminus Y$ to be $\frac{m(G)}{k}$-regular. Thus, doing this again to $(X^{'} \setminus X, Y)$ yields to a $\frac{m(G)}{k}$-regular bipartite hypergraph. We can extract a perfect matching M from such a graph. There is no edge between $(X^{'} \setminus X)$ and $(Y^{'} \setminus Y)$ therefore there are $2(n-k)$ edges of M saturating the new vertices. The other edges of M, that is to say $2n - k - 2(n-k) = k$ edges, are in G. Their set forms a valid matching P of G. Suppose that there is a vertex x of degree $\frac{m(G)}{k}$ in G. Then no edge was added to it. Therefore it is saturated by P.

□

A maximal matching of bipartite multigraph G can be found in $\mathcal{O}(n(G)^{1/2} \times m(G))$ using the Hungarian method (see [12]). Therefore, it is the complexity for finding the matching of lemma 1.

Proof of Proposition 4: If $\Delta(G) > \frac{m(G)}{k}$, add edges to the vertices of smallest degrees in order to have $\Delta(G) = \frac{m(G)}{k}$. The bound $\eta_s(G)$ doesn't change, and therefore the result too. The proof is done easily by induction on $\frac{m(G)}{k}$:

if $\frac{m(G)}{k} = 1$ then there are k independent edges and the decomposition is achieved in one step. Otherwise we subtract from G a matching of lemma 1, which gives a multigraph $G^{'}$ verifying $\frac{m(G^{'})}{k} = \frac{m(G)}{k} - 1$. Since there are at most k vertices of degree $\frac{m}{k}$, the greatest degree of $G^{'}$ has decreased, and $G^{'}$ is such that $\Delta(G^{'}) \leq \frac{m(G^{'})}{k}$. By induction we obtain a decomposition of $\eta_s(G) = \frac{m(G)}{k}$ steps.

□

This decomposition is achieved in $\mathcal{O}(n(G)^{1/2} \times m(G)^2)$.

The authors of article [4] provide a polynomial time algorithm which proves Proposition 3 for graphs, and shows that the number of steps is bounded by a polynom in $n(G)$. We use it in section 7.

We separately studied on η_s and η_d, what about η? There are quite simple graphs [13] (with all the edges having the same weight) such that η is not reached, and we can exhibit class of graphs (for instance graphs with edges having the same weight and with $k|m(G)$) for which it is.

7 Algorithms

The following algorithm approximates KPBS with a constant ratio. Before describing the algorithm, we consider a particular class of graphs such that the parameter η_s is equal to 1. Let G be a graph such that $\eta_s(G) = 1$. By definition, we have $\Delta(G) = 1$ and $m(G) \leq k$. Thus, the scheduling is composed of 1 step and the cost of this scheduling corresponds to the lower bound. For the remainder of the section, we only consider graphs G such that $\eta_s(G) \geq 2$.

In each matching of the solution the edges have the same weight, and in order to evaluate the solution, we decide that all steps have the same length α, where α is a constant which will be fixed later. The algorithm splits each edge in edges of weight α (it is an idea used in [1]) to make a multigraph H, then we find a solution such that the number of matching is minimum (thanks to Proposition 4).

Algorithm 1

Input: A weighted bipartite graph $G = (V_1, V_2, E, w)$ and an integer k
a rational number α
Output: A set of valid weighted matchings.

1. Split every edge e of G into $\lceil \frac{w(e)}{\alpha} \rceil$ edges having each a weight equal to α, which leads to a multigraph H.
2. Find $\eta_s(H)$ valid weighted matchings whose union is H.
3. Every matching represents a communication step of length α.

Complexity: Its complexity is $\mathcal{O}\left(n(H)^{1/2} \times m(H)^2\right) = \mathcal{O}\left(n(G)^{1/2} \times m(G)^2 \times W(G)^2\right)$ and therefore pseudo-polynomial since the running time of Algorithm 1 depends linearly on the weights of G.

Quality of the solution: Let $cost(G, \alpha)$ be the cost of the solution given by Algorithm 1.

Assume first, that the weights of the edges of G are multiple of α. The definitions of η_s and η_d imply $\alpha \times \eta_s(H) \leq \eta_d(G) + \alpha$ and therefore:

$$\begin{aligned} cost(G,\alpha) &= \alpha \times \eta_s(H) + \beta \times \eta_s(H) && (1)\\ &\leq \eta_d(G) + \frac{\beta}{\alpha} \times \eta_d(G) + \alpha + \beta && (2)\\ &\leq \left(\frac{\beta}{\alpha} + 1\right) \times \eta_d(G) + \alpha && (3) \end{aligned}$$

Therefore, consider the case where $\alpha = \beta$. Since only graphs G such that $\eta_s(G) \geq 2$ are considered, we have $\eta(G) \geq \eta_d(G) + 2\beta$. From equation 2, we get

$$cost(G, \beta) \leq 2\eta_d(G) + 2\beta \leq 2\eta(G) - 2\beta \qquad (4)$$

Therefore, the approximation ratio is 2 with $\alpha = \beta$.

When the weights are not multiple of α, they are rounded up to the first multiple of α, to make a graph $G^{'}$, then the previous algorithm is applied to $G^{'}$. So, from equation 2, we get

$$cost(G, \alpha) = cost(G', \alpha) \leq \eta_d(G') + \frac{\beta}{\alpha} \times \eta_d(G') + \alpha + \beta \qquad (5)$$

We compare $\eta(G)$ to $\eta(G^{'})$. We have $\eta_s(G^{'}) = \eta_s(G)$, but $\eta_d(G^{'})$ differs:

$$\begin{aligned} \eta_d(G^{'}) &= max\left(W(G^{'}), \left\lceil \frac{P(G^{'})}{k} \right\rceil\right) && (6)\\ &\leq max\left(W(G) + (\alpha-1)\Delta(G), \left\lceil \frac{P(G) + (\alpha-1)m(G)}{k} \right\rceil\right) && (7)\\ &\leq \eta_d(G) + (\alpha - 1) \times \eta_s(G) && (8) \end{aligned}$$

From in-equations 5 and 8, we get

$$cost(G, \alpha) \leq \eta_d(G')(1 + \frac{\beta}{\alpha}) + \alpha + \beta \qquad (9)$$

Consider the case where $\alpha = \beta$.

Algorithm 2

Input: A bipartite graph G.
Output: A set of valid weighted matchings.

1. Calculate $\gamma = \frac{P(G)}{k\times(n(G)^2+n(G)+1)}$
2. If $\gamma \leq \beta$, branch on Algorithm 1 with G and $\alpha = \beta$ as input
3. Otherwise, branch on the algorithm which find the valid weighted matchings such that the useful transmission time is minimized

$$\begin{aligned} cost(G,\alpha) &\leq \eta_d(G')(1+\frac{\beta}{\alpha})+\alpha+\beta && (10)\\ &\leq 2\eta(G)+2(\beta-\eta_s(G)) && (11)\end{aligned}$$

We can modify the instance of KPBS in order to have $\beta = 1$. Moreover, since we only consider graphs G such that $\eta_s(G) \geq 2$, Algorithm 1 is a pseudo-polynomial time algorithm for KPBS with an approximation ratio 2. We use now this algorithm to describe a polynomial-time algorithm for KPBS with an approximation ratio 2. Given a graph G, we evaluate an expression depending on $P(G)$ which represents roughly the average duration time of a step, then depending on the result of its comparison with the step set-up delay β, we branch on the previous algorithm or on another.

Since $\gamma \leq \beta$ all the weights of G are bounded, therefore, algorithm 1 is polynomial. Indeed $W(G) \leq P(G) \leq \beta k(n^2(G)+n(G)+1)$. This yields to a complexity of $\mathcal{O}(kn^{5/2}(G)\times m^2(G))$

We need to determine the approximation ratio in the second case (when executing line 3). The paper [4] gives (with a matrix formulation) a polynomial algorithm for optimizing the useful transmission with in the worst case a number of steps lower than $(n(G)^2+n(G)+1)$. For this algorithm, we have: ($cost(G)$ be the cost of the solution given by Algorithm 2, when executing line 3).

$$\begin{aligned} cost(G) &\leq \eta_d(G)+\beta\times(n(G)^2+n(G)+1)\\ &\leq \eta_d(G)+\frac{P(G)}{k}\\ &\leq 2\times\eta(G)\end{aligned}$$

Therefore, we can deduce that:

Theorem 3 *There is a polynomial-time approximation algorithm for KPBS with an approximation ratio* 2.

8 Heuristics

Here are two heuristics which appear to work well in practice. Unfortunately, we are at present unable to prove an upper bound for these two heuristics. They are polynomial-time algorithms because at each loop, an edge has been removed (method used in [1], [4] and [5] to obtain polynomial-time algorithms).

Complexity: We use the Hungarian method of complexity $\mathcal{O}(m(G)\times n(G)^{1/2})$ for finding a maximum cardinality matching in a bipartite graph. For both heuristics, at each step, at least one edge is removed from G. Therefore, the complexity of both heuristics is $\mathcal{O}(m(G)^2\times n(G)^{1/2})$ which is better than the complexity of algorithm 2.

Heuristic on weights

Input: A bipartite graph G.
Output: A set of valid weighted matchings.

1. Find a maximal matching.
2. Keep only the k (or less if there are less than k edges) edges whose weights are the biggest.
3. Set all the weights of the matching equal to the lowest one.
4. Subtract the matching from G.
5. Loop until there is no more edge left in G.

Heuristic on degrees

Input: A bipartite graph G.
Output: A set of valid weighted matchings.

1. Find a maximal matching.
2. Keep only the k (or less if there are less than k edges) edges with highest degrees.
3. Set all the weights of the matching equal to the lowest one.
4. Subtract the matching from G.
5. Loop until there is no more edge left in G.

Experiments: We have tested each heuristic (with n and k fixed) on a sample of 100000 random graphs (the number of edges, the edges, and finally the weights were chosen randomly with a uniform distribution). We took $\beta = 1$ and made a difference between lightly and heavily weighted graphs. Small weights were taken between 1 and 20, whereas large weights were taken between 1 and 100000. The result of an heuristic is calculated as the quotient of the cost of the solution and the lower bound η. The graphics show the average and the maximum calculated over the samples.

For these tests, the maximum is always below 2.5, even 1.8 for small weights, and the average is always below 2, and even 1.3 in case of large weights. Unfortunately, we didn't succeed into giving an approximation ratio for these two heuristics.

We explain the convex shape of the plots as follows:

- when $k = 1$ the two heuristics obtain the optimal solution which consists in one communication per steps;
- when k is greater than 2 and lower than a certain value (close to $n/2$), the quality of the solution degrades (compared to the lower bound); We believe that this is due to the fact that, at each step, the number of valid matchings increases;
- When k is greater than $n/2$ the quality of the solution tends to improve. At each stage of the two heuristics the choice of valid matchings decreases. Therefore, the heuristics are less likely to select bad valid matchings.

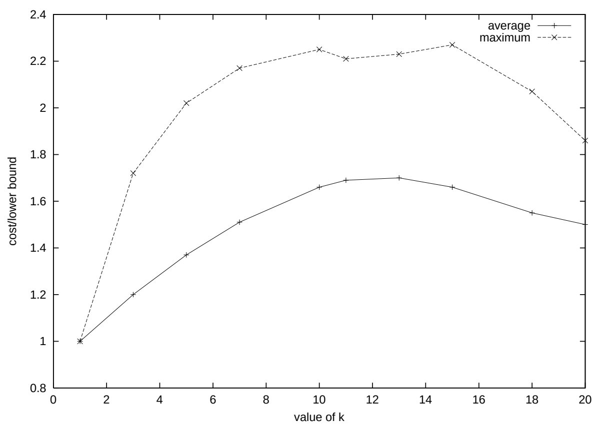

Figure 5: Heuristic on weights. $n = 20$. Simulation on 100000 graphs with small weights per point.

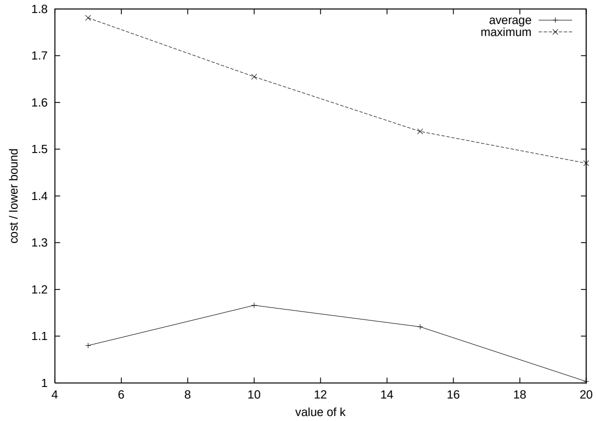

Figure 6: Heuristic on weights. $n = 20$. Simulation on 100000 graphs with large weights per point.

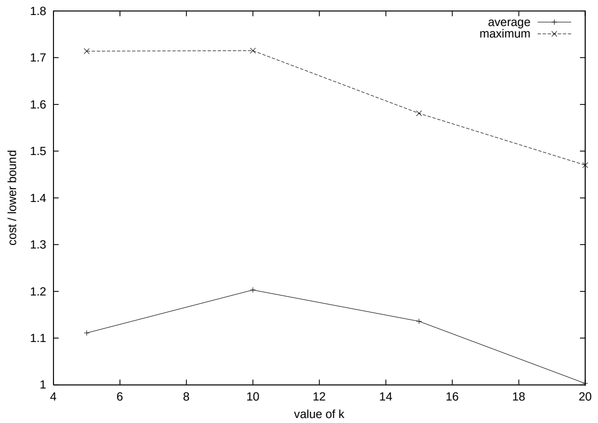

Figure 7: Heuristic on edges. $n = 20$. Simulation on 100000 graphs with large weights per point.

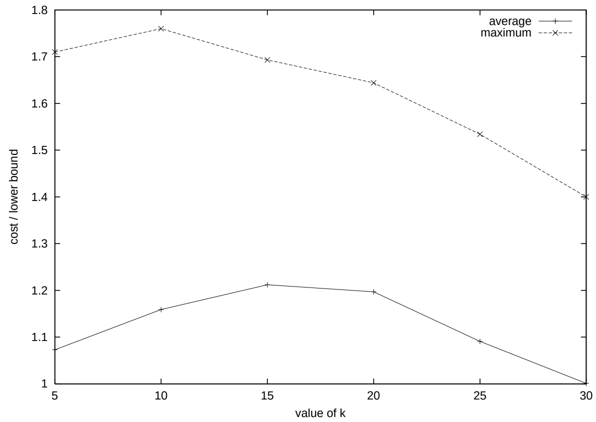

Figure 8: Heuristic on edges. $n = 30$. Simulation on 100000 graphs with large weights. per point.

9 Conclusions

In this paper we have formalised and studied the problem (called KPBS) of redistributing parallel data over a backbone. Our contribution is fourfold:

- We have shown that KPBS remains NP-Complete when k is constant,
- we have shown that approximating KPBS problem within a ratio number smaller that $\frac{4}{3}$ is impossible unless $P = NP$,
- we have studied lower bounds related to KPBS,
- we have proposed a polynomial time approximation algorithm with ratio 2,
- we have studied fast and two simple heuristics that have good properties in practice.

Our future work is directed towards studying the problem when the throughput of the backbone varies dynamically or when the redistribution pattern is not completely known in advance. We would also like to perform real tests on real architectures in order to compute a realistic value of the startup time and to be able to build a library for parallel redistribution.

References

[1] F. Afrati, T. Aslanidis, E. Bampis, and I. Milis. Scheduling in switching networks with set-up delays. In *AlgoTel 2002*, Mèze, France, May 2002.

[2] C. Berge. *Graphs*, chapter 7, pages 132–133. North-Holland, 1985.

[3] P. B. Bhat, V. .K. Prasanna, and C. .S. Raghavendra. Block Cyclic Redistribution over Heterogeneous Networks. In *11^{th} International Conference on Parallel and Distrinuted Computing Systems (PDCS 1998)*, 1998.

[4] G. Bongiovanni, D. Coppersmith, and C. K. Wong. An Optimum Time Slot Assignment Algorithm for an SS/TDMA System with Variable Number of Transponders. *IEEE Transactions on Communications*, 29(5):721–726, 1981.

[5] P. Crescenzi, D. Xiaotie, and C. H. Papadimitriou. On Approximating a Scheduling Problem. *Journal of Combinatorial Optimization*, 5:287–297, 2001.

[6] F. Desprez, J. Dongarra, A. Petitet, C. Randriamaro, and Y. Robert. Scheduling Block-Cyclic Array Redistribution. *IEEE Transaction on Parallel and Distributed Systems*, 9(2):192–205, 1998.

[7] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problem. *SIAM J. Comput.*, 5:691–703, 1976.

[8] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the theory of NP-Completeness*. a Serie of books in mathematical sciences. W.H Freeman and compagny, 1979.

[9] G. A. Geist, J. A. Kohl, and P. M. Papadopoulos. CUMULVS: Providing Fault-Tolerance, Visualization and Steering of Parallel Applications. *International Journal of High Performance Computing Applications*, 11(3):224 – 236, August 1997.

[10] I.S. Gopal and C.K. Wong. Minimizing the number of switching in an ss/tdma system. *IEEE Trans. on Communications*, 1885.

[11] Oak Ridge National Labs. Mxn. `http://www.csm.ornl.gov/cca/mxn`.

[12] S. Micali and V. V. Vazirani. An $o(\sqrt{(v)}e)$ algorithm for finding a maximum matching in general graphs. In *Proc. 21st Ann IEEE Symp. Foundations of Computer Science*, pages 17–27, 1980.

[13] N. Padoy. Redistribution de données entre deux grappes d'ordinateurs. Rapport de stage, de l'École Normale Supérieure de Lyon, 2002.

[14] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399