Partition Re-assignment: a theoretical study

Amin Mesmoudi[1], Mohand-Saïd Hacid[2], and Emmanuel Coquery[2]

[1] LIAS/ENSMA, Poitiers University
[2] LIRIS, Lyon 1 University

1 Partition Re-assignment Problem

The quality of data partitioning is mainly related to the workload. We believe that partitioning technique could be adapted to the workload. In this vein, we propose two levels of partitioning:

- An initial partitioning: In this level, partitioning depends on the kind of data. It is performed during data loading. We offer partitioning tools and we leave to the system administrator the choice of how to select the partitions.
- Partitions update: We propose another technique that allows to choose the best possible partition placement by taking into account the cost related to query processing.

In this section, we discuss the design of the technique allowing to choose the best location for partitions. From a usage point of view, we offer the system administrator a tool to analyze the transfer costs between partitions, to choose another location able to satisfy constraints on system operation and to transfer partitions from one machine to another. Doing so, two problems may arise:

1. Shutdown Duration: We have to avoid transfer of costly partitions in time. Indeed, for the update, we must shutdown the system. As the system can not stay shutdown for a long period, it is necessary to integrate a constraint on the duration of the update. We express this constraint by using the maximum size of data to be sent or received by a machine.
2. Changing partition locations can cause load balancing problems. Indeed, one can have a machine with n TB of data and another with m GB! where $n >> m$.
   To remedy to this problem, we propose to use another constraint related to the maximum difference between the data size of two machines.

Example Let illustrate our problem with an example. In a given system, data is organized in four (4) partitions P_1, P_2, P_3, P_4. The system relies on three workers M_1,M_2 and M_3. Figure 1 shows an example of partition locations and transfer logs between partitions. P_1 and P_4 are hosted by M_1, P_3 is hosted by M_2 and P_2 is hosted by M_3. A cell, in the transfer matrix, indicates the number of network packets exchanged by two partitions.

Each partition has a related network cost. This cost is the number of packets transferred between partitions that are not hosted by the same machine. For our example, this cost is equal to 2219. By changing the location of P_4 from M_1 to M_2, the cost is equal to 1480. We notice a gain of 739 packets.

	M1	M2	M3
P1	1	0	0
P2	0	0	1
P3	0	1	0
P4	1	0	0

[][Partitions location]

[][Transfer logs between partitions]

	P1	P2	P3	P4
P1	0	500	300	25
P2	20	0	50	25
P3	10	150	0	40
P4	300	100	1024	0

Fig. 1. Example of partition location and transfer logs between partitions

Complexity of the problem We could formalize partition re-assignment problem as follows:

- **Instance:**
  - P: a set of partitions
  - M: a set of machines
  - TM: Px$P \rightarrow N^{+}$, transfer logs between partitions
  - S: $P \rightarrow N^{+}$, the data size of the partition
  - X_0:$P \rightarrow M$, initial assignment of partitions
  - V_L: volume limit
  - D_{SL} : different size limit
- **Question:** find X:$P \rightarrow M$ which minimizes:

$$\sum_{p,p^{'} \in P} Y(p,p^{'})\, TM(p,p^{'}) \tag{1}$$

This formula allows to calculate the network transfers (the principal cost we consider). We consider the cost of transfer between two partitions if those partitions are not in the same machine.
with

$$Y(p,p^{'}) = \begin{cases} 0\, iff\, X(p) = X(p^{'}) \\ 1\, otherwise \end{cases} \tag{2}$$

Y indicates if two partitions are in the same machine or not.

$$\forall m \,\in\, M \sum_{p \in P} K(p,m)\, S(p) \,\leq\, V_L \tag{3}$$

This formula allows to control the volume of data sent by a machine and

$$\forall m \in M \sum_{p \in P} \overline{K}(p,m)\, S(p) \leq V_L \tag{4}$$

allows to control the volume of data received by a machine.
with

$$K(p,m) = \begin{cases} 0\, iff\, X(p) = X_0(p)\, or\, X(p) \neq m \\ 1\ otherwise \end{cases} \tag{5}$$

K indicates if a partition has been transferred to another machine

$$\forall m, m^{'} \in M\, |(\sum_{\substack{p \in P\, and \\ X(p)=m}} S(p)) - (\sum_{\substack{p^{'} \in P\, and \\ X(p^{'})=m'}} S(p^{'}))| \leq D_{SL} \tag{6}$$

With this formula, we control the difference between data sizes of the machines.

Theorem 1. *PP is NP-Hard.*

Let us start by some recalls on how to prove that a given problem is in the NP-hard class.
We say that an optimization problem A is NP-hard if: [?]

- A belongs to NP: There exists a polynomial algorithm allowing to verify that a candidate is a valid solution of the problem A.
- a decision problem $A^{'}$ related to A is NP-Complete.

Definition 1. *(Decision problem) A decision problem [?] is a problem with only two possible instances: "Yes" or "No"*

To prove that a problem $A^{'}$ is NP-Complete, we have to reduce a well known NP-complete problem to our problem, *i.e.*, all inputs of the well known NP-complete problem can be represented as special cases of the problem $A^{'}$ and a solution of $A^{'}$ is a solution of the well known NP-complete problem. The last part of the proof consists in proving the correctness of the reduction.

We start by considering PP_d decision problem related to PP. With the same inputs, the question corresponding to PP_d is the following:

Question: Given a positive integer G, is there $X{:}P \rightarrow M$ where:
$\sum_{p,p^{'} \in P} Y(p,p^{'}) \times TM(p,p^{'}) \leq$ G
Formulas 2,3,4,5 and 6 hold.

First of all, PP_d belongs to NP since checking if a placement satisfies constraints related to D_{SL}, V_L and G could be performed in polynomial time.

We subsequently propose to represent Knapsack (KS) as a special case of PP_d:

Knapsack problem:

Instance:

- O: a set of objects
- $W : O \longrightarrow Z^{+}$: weight of objects
- $V : O \longrightarrow Z^{+}$: value of objects

- w_l: weight limit of the Knapsack

Question: given an integer k, is there a subset $U \subseteq O$ such that:
$\sum_{o \in U} W(o) \leq w_l$, and
$\sum_{o \in U} V(o) \geq k$
We propose the flowing transformation (KS $<_R PP_d$):

- $M = \{m_1, m_2\}$: We consider a special case of PP_d: PP_R with 2 machines.
- $P = O \cup \{p_v\}$: partitions are objects of the knapsack problem with an additional virtual partition (p_v)
- $X_0(p_v) = m_2$, $\forall o \in O$, $X_0(o) = m_1$
- $S(o) = W(o)/o \in O$ and $S(p_v) = w_l + 1$
-

$$TM(p, p^{'}) = \begin{cases} V(p^{'})\ iff\ p = p_v\ and\ p \in U \\ 0\ otherwise \end{cases}$$

- $G = \sum_{o \in U} V(o)$ - k
- $D_{SL} = \sum_{o \in O} W(o) + S(p_v)$

It is obvious that a solution of PP_R implies $X(p_v) = m_2$

Now we will prove that each instance for KS is an instance for PP_R, for this we have to prove that:

$o \in U$ iff $X(o) \neq X_0(o)$

We start by considering that X is a solution for PP_R.

$$\sum_{o \in U} S(o) = \sum_{o \in U} S(o) = \sum_{\substack{o, X(o) \neq X_0(o) \\ and\, o \neq p_v}} S(o) \tag{7}$$

$$= \sum_{\substack{o, X(o) = m_2 \\ and\, o \neq p_v}} S(o) \tag{8}$$

$$= \sum_{\substack{o, X(o) = m_2 \neq X_0(o) \\ and o \neq p_v}} S(o) + \sum_{\substack{o, X(o) = m_1 \neq X_0(o) \\ o = p_v \\ and\, X(o) = X_0(o)}} S(o) \tag{9}$$

$$= \sum_{o} K(o, m_2) S(o) \leq V_L = w_l \tag{10}$$

$$\tag{11}$$

$$\sum_{o \in U} V(o) = \sum_{\substack{o, X(o) \neq X_0(o) \\ and\, o \neq p_v}} V(o) \tag{12}$$
$$= \sum_{\substack{o, X(o) \neq X_0(o) \\ and\, o \neq p_v}} TM(p_v, o) \tag{13}$$
$$= \sum_{\substack{o, X(o) \neq X_0(o) \\ and\, o \neq p_v}} Y(p_v, o) TM(p_v, o) \tag{14}$$
$$\tag{15}$$

We assume that:

$\vartheta = \sum_{o \in O} V(o) = \sum_{p \neq p_v} TM(p_v, p)$

$$\sum_{o \in U} V(o) = \vartheta - \sum_{o \notin U} V(o) \tag{16}$$
$$= \vartheta - \sum_{\substack{X(o) = X_0(o) \\ and\, o \neq p_v}} TM(p_v, o) \tag{17}$$
$$= \vartheta - \sum_{\substack{X(o) = X_0(o) \neq m_2 = p_v \\ and\, o \neq p_v}} Y(p_v, o) TM(p_v, o) \tag{18}$$
$$= \vartheta - \sum_{o \in O \cup \{p_v\}} Y(p_v, p) TM(p_v, o) \quad because \mathrm{Y}(\mathrm{p}_v, p) = 0 \text{ in the other cases} \tag{19}$$
$$= \vartheta - \sum_{o, o\prime \in O \cup \{p_v\}} Y(o\prime, o) TM(o\prime, o) \quad because \mathrm{TM}(\mathrm{o}\prime, o) = 0\, if\, o\prime \neq p_v \tag{20}$$
$$\geq \vartheta - G = k \quad if \sum_{o, o\prime \in O \cup \{p_v\}} Y(o\prime, o) TM(o\prime, o) \leq G \tag{21}$$

Therefore:

$G = \vartheta - k$, $\vartheta \geq k$, otherwise knapsack does not have a solution

We will show that a solution of KS is a solution of PP_R. We assume that U is solution of KS.

We have, $o \in U$ iff $X(o) = m_2$ and $X(p_v) = m_2$

$o \notin U$ iff $X(o) = m_1$

$$\sum_{o,o\prime \in O\cup\{p_v\}} Y(o\prime,o)TM(o\prime,o)$$

$$= \sum_{o,\in O} Y(p_v,o)TM(p_v,o) \quad because \text{TM}(o\prime,o) = 0\, if\, o\prime \neq p_v$$

$$= \sum_{X(p_v)\neq X(o)} TM(p_v,o)$$

$$= \sum_{X(o)\neq m_2} TM(p_v,o)$$

$$= \vartheta - \sum_{o\in U} V(o)$$

$$\leq \vartheta - k \quad \sum_{o\in U} V(o) \geq k$$

$$\leq G \quad \vartheta - k = G$$

We have to prove that $\forall m \in M \sum_{o\in O\cup p_v} K(o,m)S(o) \leq V_L$

We have two cases: $m = m_1$ and $m = m_2$

$$\sum_{o\in O\cup p_v} K(o,m_1)S(o) \quad if \text{m=m}_1$$

$$= 0 \leq V_L$$

In the second case:

$$\sum_{o\in O\cup p_v} K(o,m_2)S(o) \quad if o\in \text{U, k}(m_1,o)\text{=0 and } k(m_2,o) = 1$$

$$= \sum_{o\in U} S(o) \leq w_l = V_L \quad if o\notin \text{U, } k(m_1,o) = 0$$

$$and \quad \text{k}(\text{m}_2, 1) = 0$$

We have to prove that $\forall m \in M \sum_{o\in O\cup p_v} \overline{K}(o,m)S(o) \leq V_L$ We have two cases: $m = m_1$ and $m = m_2$

We measure received data for each machine

$$\sum_{o\in O\cup p_v} \overline{K}(o,m_1)S(o) = \sum_{o\in U} S(o) \quad if \text{m=m}_1$$

$$\leq w_l = V_L$$

In the other case:

$$\sum_{o \in O \cup p_v} \overline{K}(o, m_1) S(o) \quad \mathrm{m}_1$$
$$= 0 \le V_L$$

Last condition:
$\forall\ m, m^{'} \in M\ |(\sum_{\substack{p \in P\ and \\ X(p)=m}} S(p)) - (\sum_{\substack{p^{'} \in P\ and \\ X(p^{'})=m'}} S(p^{'}))| \le D_{SL}$
We assume that $m = m_1$ and $m^{'} = m_2$.
We recall that:

$$\sum_{o \in O} S(o) = \sum_{o \in U} S(o) + \sum_{o \notin O} S(o)$$
$$\sum_{o \in O} S(o) + S(p_v) = \sum_{o \in U} S(o) + \sum_{o \notin O} S(o) + S(p_v)$$
$$\sum_{o \in O} S(o) + S(p_v) \ge |\sum_{o \in U} S(o) - \sum_{o \notin O} S(o) - S(p_v)|$$
$$D_{SL} \ge |\sum_{o \in U} S(o) - \sum_{o \notin O} S(o) - S(p_v)|$$

If we go back to our formula:

$$|(\sum_{\substack{o \in O \cup \{p_v\}\ and \\ X(o)=m_1}} S(p)) - (\sum_{\substack{o^{'} \in O \cup \{p_v\}\ and \\ X(o^{'})=m_2}} S(o^{'}))| = |(\sum_{o \notin U} S(p)) - (\sum_{o \in U} S(o) + S(p_v)|$$
$$\le D_{SL}$$