Distribution of Indivisible Resources During Big Data Processing

Heorhii Kuchuk[1]
Andriy Kovalenko[2]

[1]Kharkiv National University of Radio Electronics, 14 Nauky Ave, Kharkiv UA-61166, Ukraine, heorhii.kuchuk@nure.ua

[2]Kharkiv National University of Radio Electronics, 14 Nauky Ave, Kharkiv UA-61166, Ukraine, andriy.kovalenko@nure.ua

Abstract. *The method of load balancing for indivisible network resources when parallelizing Big Data processing is considered. A method for finding the optimal partitioning of a processing service into parallel processes is proposed.*

Keywords: *Big Data; indivisible resource; computing resource.*

I. Introduction and Problem statement

One of the approaches used in Big Data processing is parallelization of computations [1-3]. In this case, one of the central issues in assessing the effectiveness of such processing is the analysis of processing requests to indivisible resources (IR), i.e. to those network elements that are available at any given time for no more than one request [4]. Queues to IR increase query execution time, which leads to a drop in QoS indexes. Obviously, uniform loading of IR will be the best option for increasing the speed of processing requests. Therefore, the purpose is to find such a partition of the Big Data processing process, in which the load balancing of indivisible resources is observed.

II. Problem solution and Results

Let Big Data processing service (BDPS) F uses n of IR $f_1,\dots f_n$, each of which, with one call, provides the volume of a computing resource (CR) in the amount of $r_1,\dots r_n$ units, respectively. Suppose ones can decompose F BDPS into m ($m > n$) parallel processes (PP):

$$F=\bigcup_{i=1}^{m}\left(F_i \,\middle|\, F_{i_1}\cap F_{i_2}=\varnothing \;\; \forall i_1,i_2\in\overline{1,m}\right), \tag{1}$$

$$\forall j\in\overline{1,n}\left(\exists!\,F_i \,\middle|\, F_i\supset f_j,\, i=\max_{i'}\vartheta(i',j)\right), \tag{2}$$

i.e., each PP includes no more than one IR, $\theta(i',j)$ is a frequency of f_j IR use by i' PP.

On the Cartesian product of $F\times F$ we introduce the following relation:

$$\varphi:\left(F_i,F_j\right)\to\lambda_{ij}, \tag{3}$$

where $\lambda_{ij}\left(i\neq j\right)$ is the frequency of the direct transition from F_i PP to F_j PP to use the IR located on F_j, and λ_{ii} is the frequency of requests for its own IR of F_i PP (if the corresponding resource is not requested, then $\lambda_{ij}=0$).

In such notation, the distribution of PP means finding such $\Re$ partition of F set into m disjoint (possibly empty) PPs in accordance with (1), in which the maximum load of IR is minimized:

$$\max_i \sum_{j=1}^{m}\lambda_{ij}\to\min,\quad F_i,F_j\in\Re. \tag{4}$$

Objective function (4), in contrast to the standard ones used in such situations, is focused on uniform distribution the load among all IRs. In this case, it is necessary to meet the requirements for R_c computing resource allocated for F BDPS, i.e.

$$\sum_{i=1}^{m} r_i\sum_{j=1}^{m}\lambda_{ij}\le R_c. \tag{5}$$

The formulated problem (1) - (3) for large values of m and n results in a large volume of calculations and, when applying various exact enumeration algorithms, does not provide a solution in an acceptable time. To solve it, we represent F set in the form of vertices of a fully connected graph of $G=\langle F,F\times F,\Xi,\varphi\rangle$, where the arcs of the graph are given by $F\times F$ Cartesian product; $\Xi=\{\,r_i\,\}$ is a set of vertex weights; relation φ is a set of weights of the arcs of a graph. Then the set of solutions to problem (1) - (3) is the set of all possible cuts of G graph into m subgraphs. The problem can be reduced to partitioning into m maximally internally stable subgraphs with a minimum of adjacent vertices. To determine the maximally internally stable vertices of the graph, one can use the Bron and Carbosh algorithm, which is characterized by a small computational complexity that does not rapidly increase with increasing graph dimension.

III. Conclusions

Thus, load balancing for indivisible network resources allows to speed up Big Data processing. The optimal structure of parallel processing can be constructed by finding the corresponding cuts of a fully connected graph.

References

[1] Kuchuk G., Kovalenko A., Komari I.E., Svyrydov A., Kharchenko V. Improving big data centers energy efficiency: Traffic based model and method. *Studies in Systems, Decision and Control*, vol 171. Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (Eds.). Springer Nature Switzerland AG, 2019. Pp. 161-183. DOI: http://doi.org/10.1007/978-3-030-00253-4_8

[2] Kharchenko V, Andrashov A, Sklyar V, Kovalenko A, Siora O. Gap-and-imeca-based assessment of i&C systems cyber security. *Complex Systems and Dependability*. Springer, Berlin, Heidelberg, 2013, pp. 149-164.

[3] Ruban, I., Kuchuk, H. and Kovalenko A. (2017), "Redistribution of base stations load in mobile communication networks", *Innovative technologies and scientific solutions for industries*, No 1 (1), P. 75–81, doi : https://doi.org/10.30837/2522-9818.2017.1.075.

[4] Kuchuk G., Nechausov S., Kharchenko, V. Two-stage optimization of resource allocation for hybrid cloud data store. *International Conference on Information and Digital Technologies. Zilina*, 2015. P. 266-271. DOI: http://dx.doi.org/10.1109/DT.2015.7222982