Load Balancing using Potential Functions for Hierarchical Topologies

Gy. Molnárka, N. Varjasi

Széchenyi István University
Dept. of Machine Design and Mechanics
Dept. of Information Sciences

Abstract: In this paper we consider a new approach to load balancing for parallel systems. Today's parallel computers use multiprocessors and multi-core architectures. There are big differences between homogeneous and heterogeneous parallel architectures. The new model for balancing tree topologies is based on the "amortization potential method" for homogeneous systems. Based on the our proposed model a notion can be introduced: the "goodness of the hierarchical topology", which can be characterized by potential functions. In this paper we give some examples for these potential functions, and we propose the usefulness of the model with computer experiments.

Keywords: *Parallel computing, cluster, distributed system, load balance, amortization potential*

1. Introduction

The aim of parallel programming is to break a large problem into several small components and to solve it with a concurrent system of processors. There are two main factors that play an important role in a parallel system. The first one is the amount of arithmetic calculations and the second is the communication cost between the processors and/or computational nodes. Every efficient algorithm for a given parallel system must divide evenly the computational work according to the performance capability of the system. On the other hand, communication cost must be minimal. These aspects and needs usually contradict each other. The art of parallel programming is to find a golden mean between these demands for the given parallel system.

One possible way to compromise is using the "goodness of the hierarchical topology" and its potential function.

On general systems of parallel computing the role of the computing nodes can be different. The simplest model for a parallel system is when every computing node has the same property and same role and none of them shows special behavior. The simplest parallel algorithms (for example atomic problems) could be realized with ring, hypercube or butterfly (model) topology. These topologies are effective in the case of small number of processor since the costs of communication are grow exponentially, and these topologies can only work efficiently in homogeneous systems.

Sleator [13] and Tarjan [14] examined the balancing of binary trees and give their classical analysis. Tarjan proposes the notion of the potential function as a tool for characterizing the balancing property of binary trees. Duff [6] gives a detailed analysis of parallel algorithms for linear systems of equations. In his work he focuses on the network arrangements in order to increase the efficiency of algorithms. Becciani et al. [2] define a special tree model for the N-body problem, and describe the necessity of network balancing during simulation. Lin [10] make a “load-skewing” task assignment model to minimize the delay of the cluster. He prepares a communicational model of the network burst, and the concurrent communicational conflicts and gives a good description of them. Yero [16] investigates the load balance of large master-slave systems on heterogeneous clusters. Schliephake [12] has recently developed a new design and prototype of a runtime system for parallel numerical simulations on large-scale systems.

In the following we suggest a tool for the analysis of the load balancing problem for hierarchical topology.

2. Hierarchical topology

In the case of a heterogeneous multiprocessor environment a hierarchical model can be defined. In a hierarchical model for a multiprocessor environment the processors are organized into a tree-like structure. The structure allows representing connections using parent-child relationships: each parent (master) can have many children (slaves or workers), but each child has only one parent.

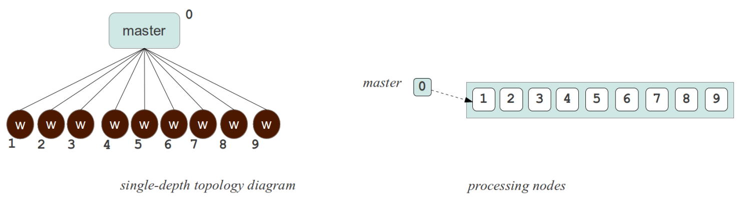

Figure 1: Single-depth hierarchical topology and its vector representation

The single-depth hierarchy is the simplest way to describe a master-slave model [16].

(See Fig. 2.) In this model the master processor has unidirectional control over other processors.

The role of the master is to divide the problem into distinct pieces, to allocate them to the corresponding slaves and finally to summarize the received results from workers. The sub-tasks are solved by the slaves.

Advantages of this system are:

– the simple hierarchy,

– the possibility to use various communication model (continuous and/or asynchronous),

– and the independence of the slave nodes, the easily convertible model.

The simplest representation of the model is a vector with the number of the nodes, where the master's number is 0 and slaves are pointed to $1, 2, \ldots, p - 1$, where p is the total number of processors.

Disadvantages of the hierarchical topology are:

– the division of the problem must be appropriate for the slaves,

– the work of the master node has a different role: to record the status of sub-tasks and it is restricted to the control of slaves

– therefore the overall efficiency can decrease [10].

Based on the principles of parallelism, (Amhdahl's, Gustaffson's law) [1] [7] the network cannot be extended effectively. By increasing the number of processors the execution time cannot be reduced beyond a certain level in this hierarchical model. The presence of the master may cause the degradation of efficiency (one less processor performs the actual work). This effect can be avoided by increasing the number of workers [4].

However, the capacity of the communication channels is limited and the number of the workers is limited also. The delay of communication service may cause starvation on worker nodes, and at the same time the master may continuously be overloaded. Such kind of systems are only effective, when the algorithmic execution time is greater than the communication time between the processors.

In the case of larger number of processors in a cluster let us examine two- or higher-level hierarchical models. These models are based on the general n-branched tree structure (see Fig. 2.). The multi-level hierarchical model serves for the load balancing of communicational channels. Between the master and slave levels one or more sub-master layer can be inserted. Communication tasks are divided between the master, sub-master and worker nodes.

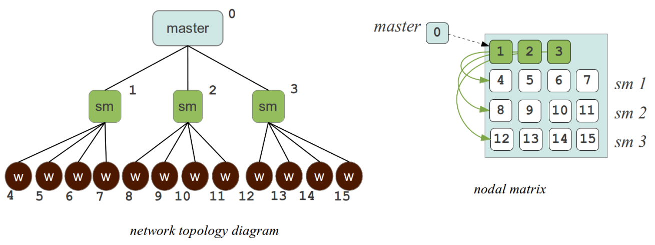

Figure 2: Tree topology and nodal matrix representation

Advantages of this topology are the better load balance and potentially lower communication costs [6]. We have to remark, that the division of the algorithms or tasks is more complicated for multi-level topologies. In the case of atomic tasks, only a small group of workers can be reached by a sub-master node [3]. By using sub-masters the overhead cost increases because of the lower number of slave processors. Such kind of topology is effective only for a large number of processors [16].

The hierarchy can be described by n-branched tree hierarchy or nodal matrix data structure (see on the right hand side of Fig. 2.).

The topology of the hierarchy [12] can be specified with the following methods:

– randomly,
– with a fixed model (in advance we define all details of the hierarchy),
– by searching the most balanced structure (this method needs a given potential function).

In the following we give the definition for the potential function definition.

3. Load balancing in homogeneous hierarchical topology

For the homogeneous topology we can define a model which is based on the results of earlier works [13], [14], [9], [5]. We use the description of binary or multiple search tree systems given by the above authors and we adopt them to the parallel computational systems with tree structures. In these papers the main result was that the optimal searching method could be realized by a balanced tree. One of the measures of the balance property of a binary tree can be described by the amortization potential function [5]. For the notation of the amortization function Φ was introduced. In the following we use the same notation, $\Phi : T \to [0,1]$, where T is the set of tree network topologies.

For parallel computational systems with tree structures the potential function can be extended in a way that the potential function describes the "strength" of the topology. The greater value of the potential function indicates a shorter execution time.

3.1. The possible form of the potential function

$$\Phi = \frac{p-1}{p};\ \Phi : N \to [0{,}1] \tag{1}$$

where p is the number of all processors (master and slaves). This potential function describes the master-slave relationship well, but does not describe the higher-level tree topologies because it does not contain information about the sub structure and the balancing property of the topology.

Let us see some generalization of the potential function (1) for the case of multi-level model.

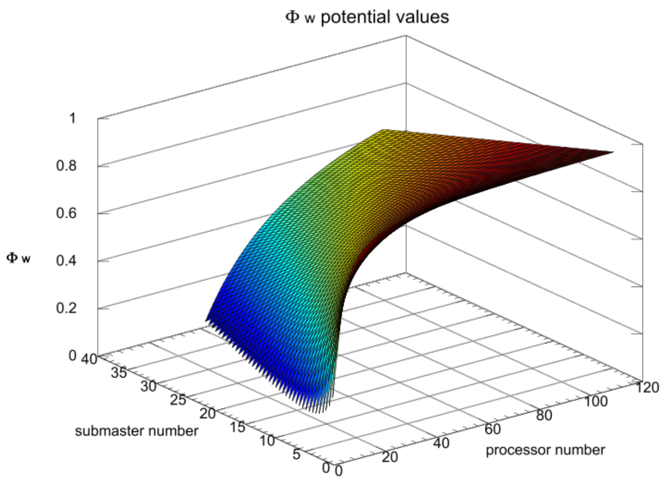

Figure 3: Potential values $\Phi_W(p,p_{sm})$

Let Φ_W be a worker potential value of the hierarchical tree topology

$$\Phi_W = \frac{p_w}{p};\ \Phi_W(p,p_w) : N \times N \to [0{,}1] \tag{2}$$

where p is the number of processors, p_{sm} is the number of sub-masters and $p_w = p - p_{sm} - 1$ is the number of worker processors. For the real hierarchical tree topology must hold that $p \geq p_w$ and $p_w \geq p_{sm}$ (see Fig. 3.1.).

The function defined by (2) describes the ratio between the worker, master and sub-master processors well, but does not give a good description about the finer details of the topology. It does not take into account how many workers are connected to a sub-master node and if the parallel communication is balanced or unbalanced.

To get a further generalization let us introduce the following Δx correction term:

$$\Delta x = \max_{i \in \{0,1,\ldots,p_{sm}\}} (sm(i)) - \min_{i \in \{0,1,\ldots,p_{sm}\}} (sm(i)), \tag{3}$$

where $sm(i)$ is the number of children of i-th sub-master node of the hierarchy. The Δx value gives the maximal difference in the number of the children of sub-masters for the whole tree.

Let us introduce the following quantity:

$$p_{avg} = \frac{\sum\limits_{i \in \{0,1,\ldots,p_{sm}\}} (sm(i))}{p_{sm}} \tag{4}$$

where p_{avg} is the average breadth of the sub-trees on the whole tree structure.

By using (3) and (4) we can define a new potential function, which describes the inhomogeneity of the tree structure:

$$\Phi_L = \frac{p_{sm} \cdot p_{avg} - \Delta x}{p_w}; \; \Phi_L : N^3 \times R \to [-1,1] \tag{5}$$

The above defined Φ_W and Φ_L functions characterize the property of the tree from two different points of view. Therefore a new potential function that contains both functions could describe the balancing property of the tree structure more adequately.

3.2. Potential values for homogeneous network

Let Φ_T be the potential value of the whole topology

$$\Phi_T = \zeta \Phi_L + (1 - \zeta)\Phi_W; \; \zeta, \Phi_T \in R[0,1], \tag{6}$$

where Φ_L and Φ_W are defined above (2), (5) and ζ is an empirical balancing value. The graph of the potential function Φ_T (see Fig. 3.2.) shows that the maximum value is attained when the tree structure is well balanced. Therefore the potential function Φ_T can be used as an indicator of the balance of the tree structure.

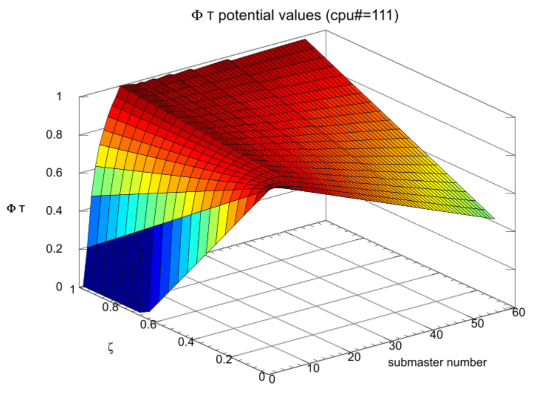

Figure 4: Potential values $\Phi_T(\zeta,p_{sm})$ with $p = 111$ nodes

Observation: For parallel cluster systems with homogeneous networks the same parallel algorithm can have different run times depending on different tree structures. Our proposed Φ_T potential function can characterize the different tree structures so that larger Φ_T value indicates shorter run time value.

More precisely: in case of the same numbers of processors the running time is shorter at a great value of Φ_T.

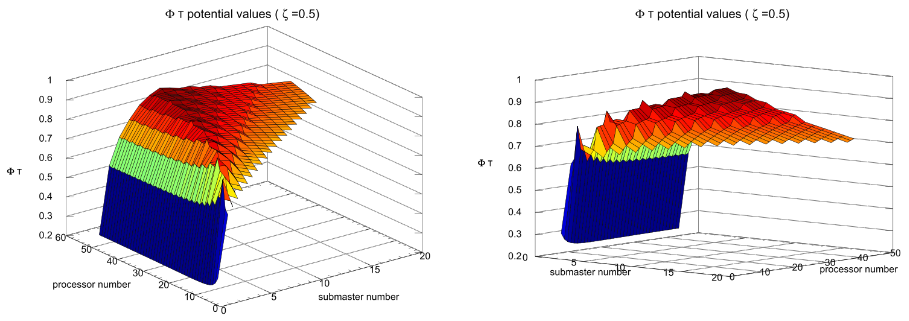

Figure 5: Potential values $\Phi_T(p,p_{sm})$ with $\zeta = 0.5$

These observations can be justified with run-time tests and computer analysis. We give some examples later in section 4.

Remark: The above formula (6) is a special potential function for homogeneous systems, but it works for heterogeneous systems as well.

For the generalization of the above proposed Φ_T potential function the general nodal matrix representation of graphs can be used. This approach could be used to define the proposed potential function for heterogeneous systems, but this approach is more sophisticated than the method we gave.

Possible methods for generalization are:

– take the difference between processor powers into consideration,

– take the communication costs between processors into consideration (speed of communication channels).

Such generalization would be very useful because the advantages of a good potential function are:

First: by using potential functions a good qualification of parallel computer network topologies can be done without benchmark tests.

Second: in the designing process of new parallel computing topologies it gives useful information. Before executing a long run-time task a "quick" test can be made to optimize the topology for a given hardware. Such experiments have been made and on the Fig. 5. shows that this idea works in practice as well.

Consequence: with the potential function (6) a well scalable hierarchy can be built.

4. Case-study for using the potential function

Let us examine some practical examples [8] [15] [11], where a simple balanced-unbalanced model can be characterized well with the potential function:

Let us see a two-level hierarchy and use $p = 13$,16 or 17 processors with different topologies, so that the potential function values will be different too. In the case of balanced trees like Fig. 4. and Fig. 4. $\Phi_T = 0.8021$ and $\Phi_T = 0.8446$. These values are better than the unbalanced case Fig. 4. with $\Phi_T = 0.5684$.

The computer tests were done by performing atomic tasks. Every worker executed the Miller–Rabin prime tests. In this case the algorithmic cost was always much higher than the communication costs (in this case our model is adequate).

Our potential function method must work for higher numbers of processors that we gave in examples Fig. 4, 4, 4.

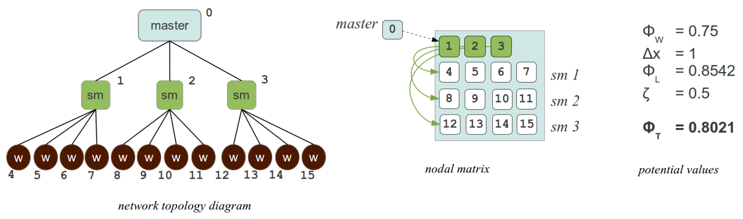

Figure 6: Load balanced topology $p = 16$, $p_{sm} = 3$

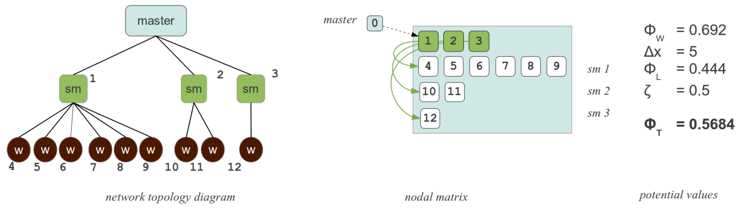

Figure 7: Unbalanced topology $p = 13$, $p_{sm} = 3$

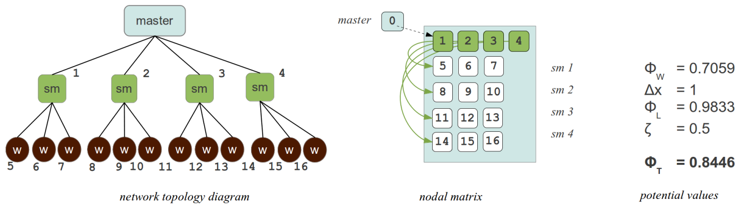

Figure 8: Balanced topology $p = 17$, $p_{sm} = 4$

On Fig. 4. we demonstrate the results of tests made on a *HP Blade c3000* system. The processor number was $p = 73$ with different sub-master number from $sm = 2$ up to $sm = 36$. in these tests we used as uniformly distributed workers as possible because the total number of possible topologies is huge (inhomogeneity was between the sub-master and worker processor numbers). With the relatively small sample the test results shows very good accordance between the expectations and computational results.

Even though this experiment does not prove the idea that our potential function is the

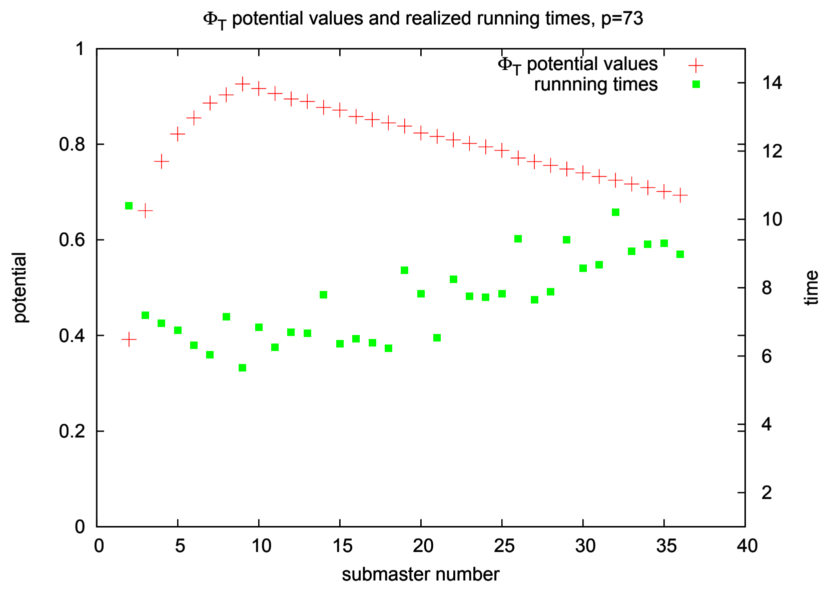

Figure 9: Topology potential values and the computer test results $p = 73$, $p_{sm} = 2..36$, $\zeta = 0.5$

appropriate indicator of the goodness of parallel network systems, but it validates it's usefulness. Therefore further research in this field is promising. The two main directions seem to be beneficial to realize more precise results: to perform further computer experiments and to formulate a more general potential function for inhomogeneous parallel computer systems.

Acknowledgements

This work was supported by TÁMOP-4.2.2-08/1/2008-0021 and by GOP-1.1.2-07/1-2008-0003 projects.

References

[1] G. M. Amdahl: *Validity of the single-processor approach to achieving large scale computing capabilities*, in Proceedings of AFIPS Conference, 1967.

[2] U. Becciani, R. Ansaloni, V. Antonuccio-Delogu, G. Erbacci, M. Gambera, A. Pagliaro *A parallel tree code for large N-body simulation: dynamic load balance and data distribution on a CRAY T3D system* Computer Physics Communications 106., 1997. pp. 105–113

[3] Y. Chen, Y. Deng: *A detailed analysis of communication load balance on BlueGene supercomputer*, Computer Physics Communications 180., 2009. pp. 1251–1258

[4] X. H. Sun, Y. Chen: *Reevaluating Amdahl's law in the multicore era* J. Parallel Distrib Comput. 70., 2010. pp. 183–188

[5] T. H. Cormen, Ch. Leiserson, R. Rivest: *Algoritmusok* Műszaki Könyvkiadó, Budapest, 2001. ISBN 963-16-3029-3

[6] I. S. Duff, H. A. van der Vorst: *Developments and trends in the parallel solution of linear systems*, Parallel Computing 25., 1999. pp. 1931–1970

[7] J. L. Gustafson: *Reevaluating Amdahl's law*, in Communications of ACM, 1988.

[8] G. Kanti: *Hierarchikus szervezésű párhuzamos programozási modellek MPI rendszerben*, Széchenyi István Egyetem Győr, 2011. p.63

[9] Z. Király: *Adatstruktúrák*, ELTE, 2011. p.92; http://www.cs.elte.hu/ kiraly/Adatstrukturak.pdf

[10] W.-M. Lin, W. Xie: *Load-skewing task assignment to minimize communication conflicts on network of workstations*, Parallel Computing 26., 2000. pp. 179–197

[11] G. Molnárka, N. Varjasi: *A simultaneous solution for general linear equations on a ring or hierarchical cluster*, Acta Technica Jaurinensis Vol. 3. No. 1., 2010. pp. 65–73

[12] M. Schliephake, X. Aguilar, E. Laure: *Design and Implementation of a Runtime System for Parallel Numerical Simulations on Large-Scale Clusters* Procedia Computer Science 4, 2011. pp. 2105–2114

[13] D.D.Sleator, R. E. Tarjan *Self-Adjusting Binary Search Trees* Journal of the Association for Computing Machinery. Vol. 32, No. 3., 1985. pp. 652–686

[14] R. E. Tarjan *Amortized Computational Complexity* SIAM Journal on Algebraic and Discrete Methods, Vol. 6. No. 2. 1985. pp. 306–317

[15] N. Varjasi: *Parallel Algorithm for linear equations with different network topologies*, Proceedings of International e-Conference on Computer Science (IeCCS) 2006 in Lecture Series on Computer and Computational Sciences, Brill Academic Publishers, 2007. pp. 502–505, ISBN 978-90-04-15592-3

[16] E. J. H. Yero, M. A. A Henriques: *Speedup and scalability analysis of Master-Slave applications on large heterogeneous clusters*, Journal of Parallel and Distributed Computing 67. 2007. pp. 1155–1167