Proceedings of the 10th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'10), 20–23 October 2010, Riga, Latvia, p. 218-224. ISBN 978-9984-818-34-4
Transport and Telecommunication Institute, Lomonosova 1, LV-1019, Riga, Latvia

COMPUTING PARALLELIZATION EFFICIENCY ESTIMATION IN THE INTELLIGENT TRANSPORTATION SYSTEMS

Boris Tsilker[1], Sergey Orlov[2]

Transport and Telecommunication Institute
Lomonosova 1. Riga, LV-1019, Latvia
E-mail: tsilker@tsi.lv[1], sorlov@tsi.lv[2]

Considerable achievements in computing and telecommunication area make possible in a new way solve a wide spectrum of transportation problems, affected in the concept of intelligent transportation systems (ITS). In a technical aspect such sort systems represent a set of interacting computational nodes, equipped with various sensors, and can be treated as distributed computer systems. The distributed structure of ITS supposes parallelization of solvable transportation tasks and their distributed realization.

The concept of efficiency of parallelized calculations presupposes a few aspects. Three such aspects are picked out in the work: calculation speed, efficiency of system scaling, and efficiency of parallel computations as compared to sequential ones. Proper metrics for each group are offered, allowing numerical characterization of certain aspect of parallelized computations. As such metrics the index of parallelism and speedup (*PI* and *S*), efficiency and utilization (*E* and *U*), redundancy and compression (*R* and *C*) are examined.

The peculiarity of intelligent transportation systems is in, that a large volume of communications is typical for them, substantially telling on the indexes of the system. Influence of communication overheads on the total efficiency of the system is analysed in the work.

The question of private indexes integration into a single integral index, characterizing the quality of ITS realization, is examined. Applicability of the suggested metrics and their evidence is illustrated by examples.

***Keywords**: intelligent transportation system, parallelization, distributed systems, efficiency, performance metrics*

1. Introduction

Information technologies (IT) have transformed many industries. In respect to transportation systems such transformations started in the past decade and are now in the early stages. IT enables elements within the transportation system — vehicles, roads, traffic lights, message signs, etc. — to become intelligent by embedding them with microchips and sensors and empowering them to communicate with each other through wireless technologies. Transportation systems realizing this approach are known as intelligent transportation systems (ITS). According to definition of the Intelligent Transportation Systems Society ITS are those utilizing synergistic technologies and systems engineering concepts to develop and improve transportation systems of all kinds.

Through the use of advanced computing, control, and communication technologies, ITS promises to greatly improve the efficiency and safety of the existing surface transportation system and reduce the transportation-related energy consumption and negative environmental impact. ITS encompass a broad range of wireless and wire line communications-based information and electronics technologies. Depending on nature of the solvable transportation tasks, ITS is made up of 20 types of technology-based systems. As an example a few categories of transportation tasks, being the objects of ITS, can be mentioned:

1. Traveller Information Services (e.g. traveller advisory systems, etc.)
2. Traffic Management Services (e.g. advanced traffic signal systems, freeway incident detection and management systems, etc.)
3. Public Transport Services (e.g. electronic transit schedule information, GPS tracking of bus movements and locations, etc.)
4. Commercial Vehicle Operations (e.g. weigh-in-motion, electronic truck clearance at vehicle inspection stations and border crossings, etc.)
5. Electronic Payment Services (e.g. electronic toll payment, transit fare payment, etc.)
6. Emergency Management Services (e.g. improving emergency vehicle response time by fleet tracking, route guidance and signal pre-emption, etc.)
7. Vehicle Safety and Control Systems (e.g. in-vehicle technologies such as on-board computers, collision avoidance sensor technologies, etc.)
8. Information Warehousing Services (e.g. traffic safety data collection, archived data management, etc.)

In less detail all ITS types are divided into intelligent infrastructure systems and intelligent vehicle systems.

2. Specificity of Parallel and Distributed Computing

The variety of transportation tasks suppose the variety of methods for construction of proper ITS. Accordingly the requirements to ITS hardware and software architecture also differ, but in any case any system must be maximally effective both in economic and technical plan. Naturally, that for the estimation of technical-and-economic efficiency it is necessary to define the most suitable criteria and system of indexes, numerically characterizing the degree of ITS compliance with these criteria.

From the presented list of transportation tasks the majority on their nature suppose distributed solution, where outcome is reached due to cooperative processing of information, incoming from numerous sources, for example, from motion sensors. This information processing, as a rule, is also decentralized. By other words, such systems represent distributed computer systems. Only limited range of ITS is realizable by centralized computer systems, but even here, by reason of high performance requirements, parallel computer systems are usually used. Thus, examining the questions of ITS efficiency estimation, it is necessary to talk about the parallel and distributed computer systems.

The problem of parallel systems efficiency estimation more than enough studied [1-3]. By now the harmonious system of indexes, which is fully applicable to ITS of this type, is produced. On the other hand, the system for estimation of the distributed systems at the present time does not exist. The attempts of efficiency estimation for such systems usually come to the separate estimation of distributed systems' components and getting some integral estimation on the base of these particular appraisals. Besides of that sort approach realization complexity, it imperfectly takes into account the impact of ITS components interaction on the total system efficiency.

We offer another approach to ITS of distributed type efficiency estimation, based on the fact that the distributed calculations are the special case of parallel calculations. At distributed calculations the current task is actually segmented, i.e. is partitioned on subtasks, which can be computed concurrently. The main property of distributed computing is in extra costs, arising up at interaction of system constituents. Because of ideological closeness of parallel and distributed calculations it is suggested for the efficiency estimation take advantage of the system of indexes, used for the parallel systems, mapping them on the distributed systems.

3. Overheads in Parallel and Distributed Computations

Overheads in parallel and distributed computations fall in three groups:

- interprocess communications (typically the most significant overhead in distributed computations);
- idling (processor may become idle because of load imbalance, synchronization, and presence of serial computation);
- excess computations (difference in computation fulfilled by the parallel program and the best sequential program is the excess computation overhead incurred by the parallel program).

Parallel overhead is encapsulated into a single expression referred to as the overhead function. Overhead function (or total overhead), T_0, of a parallel system is defined as the total time collectively spent by all n processing elements over and above time $T(1)$ required by the fastest known serial algorithm for solving the same problem on a single processing element:

$$T_0 = n \times T(n) - T(1) .$$

4. Communication Overhead in Distributed Computing

For determining communication time between elements of distributed computations t_{comm} three parameters usually are used:

- Start-up time (t_s): The time required to handle a message at the sending processor including the time to prepare the message, the time to execute the routing algorithm, and the time to establish an interface between the local processor and router.
- Per-hop time (t_h): The time for message header to travel between two directly connected processors. Also known as node latency.
- Per-word transfer time (t_w): The time for a word traverse a link. If the channel bandwidth is r words per second, then per-word transfer time is $t_w = \frac{1}{r}$.

Thus, $t_{comm} = t_s + t_h + t_w$.

Consider two routing modes: store-and-forward routing and cut-through routing.

At store-and-forward routing a message is traversing a path with multiple links: each intermediate processor on the path forwards the message to the next processor after it has received and stored the entire message. In that case the communication overhead can be described by expression

$$t_{comm} = t_s + (mt_w + t_h)l \ ,$$

where m — message size in words, l — path length in channel number.

Usually $t_h << mt_w$, therefore the expression is simplified to

$$t_{comm} = t_s + mt_w l \ .$$

At cut-through routing a message is forwarded at intermediate node without waiting for entire message to arrive with no buffering in memory (busy link causes worm to stall; deadlock may ensure). Here the communication overheads are represented by expression

$$t_{comm} = t_s + mt_w + lt_h \ .$$

Again, considering t_h, to be small compared to mt_w, the communication overhead is

$$t_{comm} = t_s + mt_w \ .$$

5. Parallel Computing Metrics

Parallel computing metrics are a system of indexes, allowing estimate the advantages, got at task solution concurrently on n processors, as compared to the serial solution of the same task on single processor. On the other hand, they permit to judge about the validity of usage of given processor number for solving of concrete task. Under *parallel computing* will understand the sequence of steps, where each step consists of i operations, implemented simultaneously by a set of i processors, working in parallel. The basis for definition of the mentioned metrics form followings characteristics:

- n — number of processors, used for organization of parallel calculations;
- $O(n)$ — amount of calculations, expressed through the number of operations, executable by n processors during the task solution;
- $T(n)$ — total time of calculations (task solution) with the use of n processors.

We will arrange, that time changes discretely, and a processor executes any operation in time slice. The followings correlations are hereupon valid for time and volume of calculations: $T(1)= O(1)$, $T(n) \leq O(n)$. The last correlation formulates assertion: *time of calculations can be shortened due to distributing of calculation volume on a few processors*.

The number of processors used by a program at a particular point in time defines the *degree of parallelism* $P(t)$. The plot of parameter P against time is called the *parallelism profile* for the program. Changes in the $P(t)$ depend on many factors (algorithm, available resources, compiler optimisations, etc.). A typical parallelism profile for a *divide-and-conquer* algorithm is shown in Fig. 1.

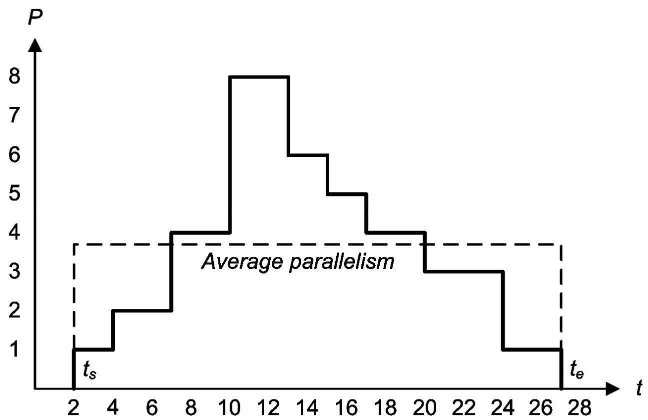

Figure 1. Parallelism profile for typical divide-and-conquer algorithm

Suppose that system consists of *n* of homogeneous processors. Let's express the computing capacity of a single processor Δ. as an amount of operations in a time unit, not taking into account overheads, related to memory access and data transmission. If for observed period of time (certain amount of time slices) are loaded *i* processors, then $P = i$. Thus, a program runs from the start time t_s to the moment of completion t_e needs a total of $O(n)$ operations. $O(n)$ is proportional to area under to the curve of parallelism profile:

$$O(n) = \Delta \sum_{i=1}^{n} i t_i ,$$

where t_i is a time interval (common amount of time slices), during which $P = i$, and $\sum_{i=1}^{n} t_i = t_e - t_s$ is common time of calculations.

Average parallelism A is defined as

$$A = \frac{\sum_{i=1}^{n} i t_i}{\sum_{i=1}^{n} t_i} .$$

The parallelism profile on the Figure in course of observation time (t_s, t_e) grows from 1 to peak value $n = 8$, then go down to 0. Average parallelism $A = (1 \times 5 + 2 \times 3 + 3 \times 4 + 4 \times 6 + 5 \times 2 + 6 \times 2 + 8 \times 3) / (5 + 3 + 4 + 6 + 2 + 2 + 0 + 3) = 93/25 = 3{,}72$.

In essence it is possible to single out four groups of metrics.

The first group characterizes speed of calculations. This group is presented by a pair of metrics – *parallel index* and *speedup*.

Parallel Index characterizes average speed of parallel calculations through the amount of the executed operations:

$$PI(n) = \frac{O(n)}{T(n)} .$$

Speedup due to program concurrent execution serves as the index of effective speed of calculations. It expresses how much the performance improves when compared to the sequential execution (relative benefit). Speedup is calculated as the ratio of runtime for solving a problem on a single processor (using the best sequential algorithm) to the time taken for the same problem by a parallel system of *n* processors (at using the best parallel algorithm).

$$S(n) = \frac{T(1)}{T(n)} .$$

Remark in relation to the algorithms of task solution algorithms underline the facts that different algorithms can the best appear for serial and parallel realization, and at the estimation of the speedup it is necessary to come exactly from the best algorithms. If the fastest sequential algorithm is not known, the fastest practical approach usually is used.

The second group is formed by *efficiency* and *utilization* metrics, making possible to judge about the efficiency of bringing to the task solution of additional processors.

Efficiency characterizes the reasonability of processor number increasing through the fraction of speedup, attained due to parallel calculations, which falls on one processor:

$$E(n) = \frac{S(n)}{n} = \frac{T(1)}{nT(n)} .$$

Efficiency measures the fraction of time the processor is usefully employed.

Utilization takes into account contribution of every processor at a parallel computing, expressed as amount of operations, executed by a processor in time unit.

$$U(n) = \frac{O(n)}{nT(n)} .$$

The third group of metrics, — *redundancy* and *compression*, — characterizes efficiency of parallel computing by comparison of volume of calculations, executed at the parallel and serial task solution.

Redundancy is a ratio of parallel calculations volume and equivalent successive calculations volume:

$$R(n) = \frac{O(n)}{O(1)}.$$

It shows the extent of the workload increase for going from serial to parallel execution. Importance of this metrics is in that it will proceed not from the relative speedup and efficiency indexes, got from calculation time, but from the absolute indexes, being based on the volume of the executed computational work. Redundancy reflects the measure of agreement between software and hardware parallelism.

Note that utilization can be expressed through the redundancy and efficiency metrics:

$$U(n) = \frac{O(n)}{nT(n)} = R(n)E(n).$$

Correlation $T(1) = O(1)$ is taken into account here.

Compression is calculated as a reciprocal value of redundancy:

$$C(n) = \frac{O(1)}{O(n)}.$$

Finally, the fourth group is formed by the only metrics – *quality*, joining three considered groups of metrics.

Quality is defined as:

$$Q(n) = \frac{T^3(1)}{nT^2(n)O(n)} = S(n)E(n)C(n).$$

As far as this metrics ties up speedup, efficiency and compression metrics, it is more objective index of performance improvement due to parallel calculations, and can be considered as an common measure (integral index) defining the whole system performance.

For an example will define the numerical values of metrics in respect to the task, used for parallelism profile concept illustration (Fig. 1). Supposing that the best algorithm for a successive and parallel calculation match, have: $n = 8$; $T(1) = O(1) = O(8) = 93$; $T(8) = 25$. Then:

$$PI(8) = \frac{O(8)}{T(8)} = \frac{93}{25} = 3{,}72; \qquad S(8) = \frac{T(1)}{T(8)} = \frac{93}{25} = 3{,}72;$$

$$E(8) = \frac{T(1)}{8T(8)} = \frac{93}{8\times 25} = 0{,}465; \quad U(8) = \frac{O(8)}{8T(8)} = \frac{93}{8\times 25} = 0{,}465;$$

$$R(8) = \frac{O(8)}{O(1)} = \frac{93}{93} = 1; \qquad C(8) = \frac{O(1)}{O(8)} = \frac{93}{93} = 1;$$

$$Q(8) == S(8)E(8)C(8) = 3{,}72\times 0{,}465\times 1 = 1{,}73.$$

At completion note, that for the considered metrics the following correlations are true:

$$1 \le S(n) \le PI(n) \le n; \quad 1 \le R(n) \le \frac{1}{E(n)} \le n; \quad \frac{1}{n} \le E(n) \le C(n) \le 1;$$

$$\frac{1}{n} \le E(n) \le U(n) \le 1; \quad Q(n) \le S(n) \le PI(n) \le n.$$

6. Distributed Computing Metrics

This group of metrics is based on the parallel computing metrics, but additionally takes into account the of communications overheads of the distributed systems.

In particular, the speedup metrics goes over:

$$S_d(n) = \frac{T(1)}{T(n) + \sum_{i=1}^{n} t_{comm}(i)},$$

where $t_{comm}(i)$ is communications overhead of i-th processor, $\sum_{i=1}^{n} t_{comm}(i)$ is total communication overhead of all n processors of the distributed system.

Accordingly the efficiency and utilization metrics are written down in a form of:

$$E_d(n) = \frac{T(1)}{n \times T(n) + \sum_{i=1}^{n} t_{comm}(i)},$$

$$U_d(n) = \frac{O(n)}{n \times T(n) + \sum_{i=1}^{n} t_{comm}(i)}.$$

Formulas of integral index *of quality* are modified as follows:

$$Q_d(n) = \frac{T^3(1)}{(T(n) + \sum_{i=1}^{n} t_{comm}(i)) \times (n \times T(n) + \sum_{i=1}^{n} t_{comm}(i)) \times O(n)}.$$

Expressions for the rest metrics remain unchanged.

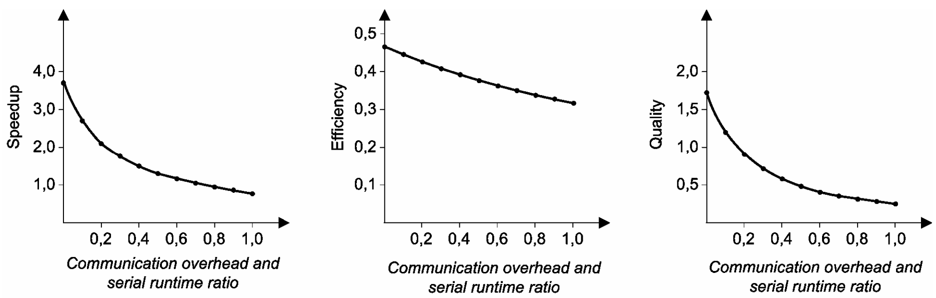

Figure 2. Efficiency indexes of a distributed system, realizing the divide and concur algorithm

Figure 2 illustrates influence of communication overhead on the efficiency indexes of a computer system at distributed solution of the before considered divide and concur algorithm. Value on a vertical axis corresponds to absence of communication overhead, i.e. to the parallel system.

7. Conclusions

The issue of the day in intelligent transportation systems development still is a task of effective selection and design of the used software and hardware tools. Within the framework of decision of this task, authors has analysed characteristics of given problem domain, singularities of structural organization and functioning for this family of computer systems, has explored their specificity, has offered a set of metrics for the estimation of efficiency of parallel and distributed modes of operation. These metrics allow estimation of integral efficiency of intelligent transportation systems, and taking into account the basic features of both parallel and distributed computing.

References

1. Voevodin, V.V., Voevodin, Vl.V. *Parallel Computing*. SPb.: BHV-Petersburg, 2002. 608 p.
2. Orlov, S.A., Tsilker, B.J. *Computer and Systems Organization,* 2nd ed. SPb.: Peter, 2010. 688 p.
3. Zomaya, Albert Y.H. *Parallel and Distributed Computing: Handbook. Series on Computing Engineering.* New York: McGraw-Hill, 1996.