Microsoft Research

2012 Summer School on Concurrency

August 22–29, 2012 | St. Petersburg, Russia

Introduction to Parallel Programming

Section 2.

Communication Complexity Analysis of Parallel Algorithms

Victor Gergel, Professor, D.Sc.
Lobachevsky State University of Nizhni Novgorod (UNN)

Contents

Introduction

- In this section there is studied how to analyze the complexity of communication operations, which occur at the time of parallel algorithm execution. This section
  - gives an overview of communication techniques,
  - considers the complexity of the basic communication operations,
  - describes methods of logical presentation of communication network topology.

Time delays, which occur in the process of data communication between processors, may be crucial for parallel computation efficiency

Overview of Communication Techniques…

- The **routing algorithms** determine the route of data communication from the sending processor to the processor, which should receive the message:
  - the *optimal* ones, which always determine the shortest path for data communication, and *non-optimal* routing algorithms,
  - *deterministic and adaptive* methods of choosing routes (the adaptive algorithms determine the route of data communication depending on the available load of communication channels)

Overview of Communication Techniques…

❑ Routing Algorithms:

- *Dimension-ordered routing* is one of the widely used optimal algorithms:
  - Searching the data communication routes is carried out for each communication network dimension in turn,
  - For a two-dimensional grid: the data is passed first in one direction (for instance, horizontally), and then is transmitted along the other direction (*XY-routing algorithm*),
  - For a hypercube: cyclic data transmission to the processor, which is determined by the first differing bit position in the numbers of those processor, who stores the message at the moment, and the processor, who has to receive the message

Overview of Communication Techniques…

- **Data Communication Methods…**

  The time necessary for transmitting data between the processors defines the *communication overheads* of the duration of parallel algorithm execution.

  The basic set of parameters, which describe the data communication time, consists of the following values:

  - ***Startup cost*** (t_s) characterizes the duration of preparing the message for transmission, the search of the route in the network, etc. This time is usually referred as the *latency* of communication operations,
  - ***Per-header transfer time*** (t_h) gives the time to transmit control data between two neighboring processors (i.e. the processors, connected by a physical data communication channel); the data header can enclose system information, the error detection data block, etc.,
  - ***Per-byte transfer time*** (t_b) is the time to transmit one byte along a data communication channel; the duration of this transmission is defined by the communication channel *bandwidth*.

Overview of Communication Techniques…

❑ Data Communication Methods…

Store-and-forward routing (*SFR*) method transmits messages as indivisible information blocks:

- the processor, which stores a message for transmission, gets all the amount of data ready for transmission, determines the processor, which should receive the data, and initializes the operation of data communication,
- the processor, to which the message has been sent, first receives all the transmitted data and only then begins to send the received message further along the route.

Overview of Communication Techniques…

❑ Data Communication Methods…

Cut-through routing (*CTR*) method is based on presenting the transmitted messages as information blocks (*packets*) of smaller sizes:

– the receiving processor may send the data further along the route immediately after receiving the next packet without waiting for the termination of the whole message data communication.

Overview of Communication Techniques…

❑ Data Communication Methods…

The time of data communication ***t*** for the message of ***m*** bytes along the route of ***l*** length is defined by the expression:

– **Store-and-forward routing**:

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

If the messages are long enough, the control data communication time may be neglected:

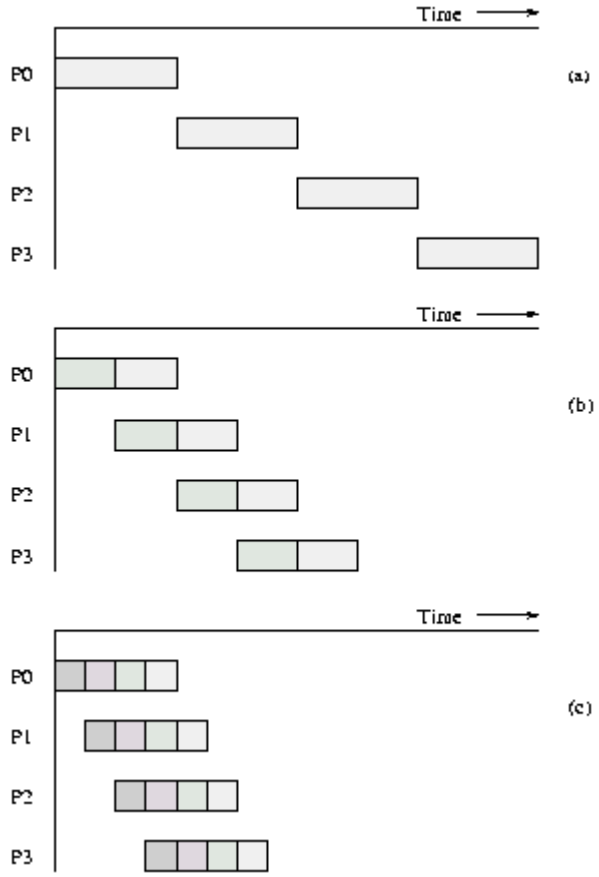
$$t_{comm} = t_s + mt_b l,$$

– **Cut-through routing**:

$$t_{comm} = t_s + mt_b + t_h l.$$

Overview of Communication Techniques…

❑ Data Communication Methods



- ❑ Store-and-forward routing
- ❑ Cut-through routing (the message is divided into 2 packets)
- ❑ Cut-through routing (the message is divided into 4 packets)

Overview of Communication Techniques

- **Cut-Through Routing** (recommendations):
  - In the majority of cases the packet communication leads to faster data communication,
  - This method decreases the need for memory for storing the transmitted data in order to arrange the message transmission,
  - Different communication channels may be used for the packet communication simultaneously,
  - The implementation of the packet communication requires the development of more complex hardware and software,
  - It may also increase the overhead expenses (initialization time and control data communication time),
  - Deadlocks may also occur in case of packet communication.

Complexity Analysis of Communication Operations…

- Certain procedures of network processor interactions may be referred to *the basic communication operations*. Such operations are either widely used in parallel computations or other message send-receive operations may be reduced to them
- Many of basic communication operations have *dual* ones which are performed by reversing the direction and sequence of transmitted data in the original operations (for instance, the operation of data communication from a processor to all the available network processors corresponds to the operation of message receiving in one processor from all the rest processors)

Complexity Analysis of Communication Operations…

- **Data Communication between Two Network Processors**

  The complexity of this communication operation can be obtained by means of substitution of the maximum path into the expression for data communication time in case of various communication methods

Topology	Store-and-Forward Routing	Cut-Through Routing
Ring	$t_s + mt_b \lfloor p/2 \rfloor$	$t_s + mt_b + t_h \lfloor p/2 \rfloor$
Grid-torus	$t_s + 2mt_d \lfloor \sqrt{p}/2 \rfloor$	$t_s + mt_b + 2t_h \lfloor \sqrt{p}/2 \rfloor$
Hypercube	$t_s + mt_b \log_2 p$	$t_s + mt_b + t_h \log_2 p$

Complexity Analysis of Communication Operations…

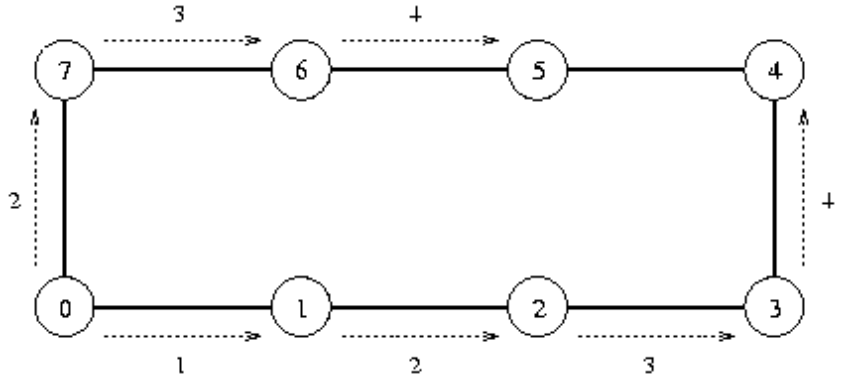
❑ One-to-All Broadcast…

One-to-all broadcast or *single-node broadcast* is one of the most widely used communication operations. *Single-node accumulation* consists in receiving the messages by one of the processors from all the other network processors, it is an operation reverse in its action to the single-node broadcast operation

Complexity Analysis of Communication Operations…

- **One-to-All Broadcast (*Store-and-forward routing*)…**

  In case of the **ring** topology the sending processor can initiate data transmission to two neighbors at once. These processors in their turn send the message further in the ring:

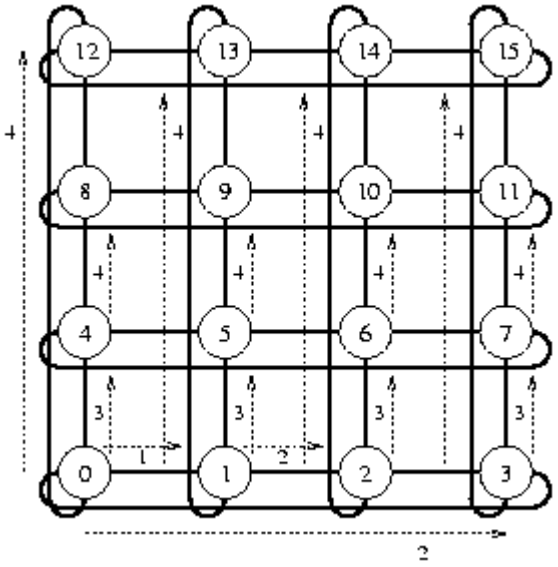


The communication time of the operation execution in this case will be defined by the following relation:

$$t_{comm} = (t_s + mt_b)\lceil p/2 \rceil$$

Complexity Analysis of Communication Operations…

❑ One-to-All Broadcast (*store-and-forward routing*)…



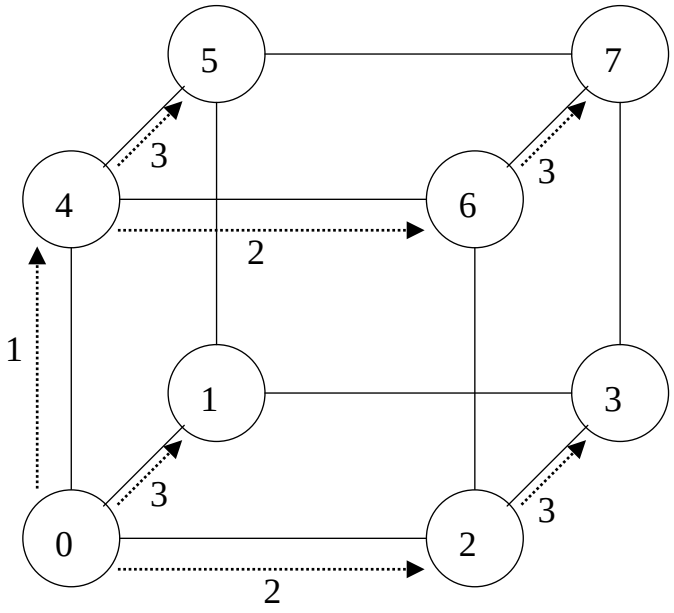
For the **grid-torus topology** broadcasting can be carried out as a two-stage procedure. At *the first stage* data are transmitted to all the processors of the network, which are located on the same horizontal line of the grid as the sending processor. During *the second stage* the processors, which received the data copy at the first stage, send the messages along the corresponding vertical lines.

The estimation of broadcasting duration in accordance to the described algorithm, is determined by the following relation:

$$t_{comm} = 2(t_s + mt_b)\lceil \sqrt{p}/2 \rceil$$

Complexity Analysis of Communication Operations…

❑ One-to-All Broadcast (*store-and-forward routing*)



For the **hypercube** broadcasting can be carried out as an N-stage data communication procedure. During *the first stage* the sending processor sends data to one of the neighbors. As a result, there are two processors, which have the copies of the data after the first stage. At *the second stage* the two processors engaged at the first stage send messages to their neighbors along the second dimension, etc.

As a result of this broadcasting the operation execution time is estimated by the following expression:

$$t_{comm} = (t_s + mt_b)\log_2 p$$

Complexity Analysis of Communication Operations…

❑ One-to-All Broadcast (*cut-through routing*)…

The broadcast algorithm for the **ring** topology can be obtained by means of logical presentation of the ring structure as a hypercube. The sending processor sends the data to the processor, which is at ***p/2*** distance from the initial processor during the broadcast stage. Further, during the second stage the two processors, which already have the data after the first stage, transmit the data to the processors, which are located at ***p/4*** distance, etc.

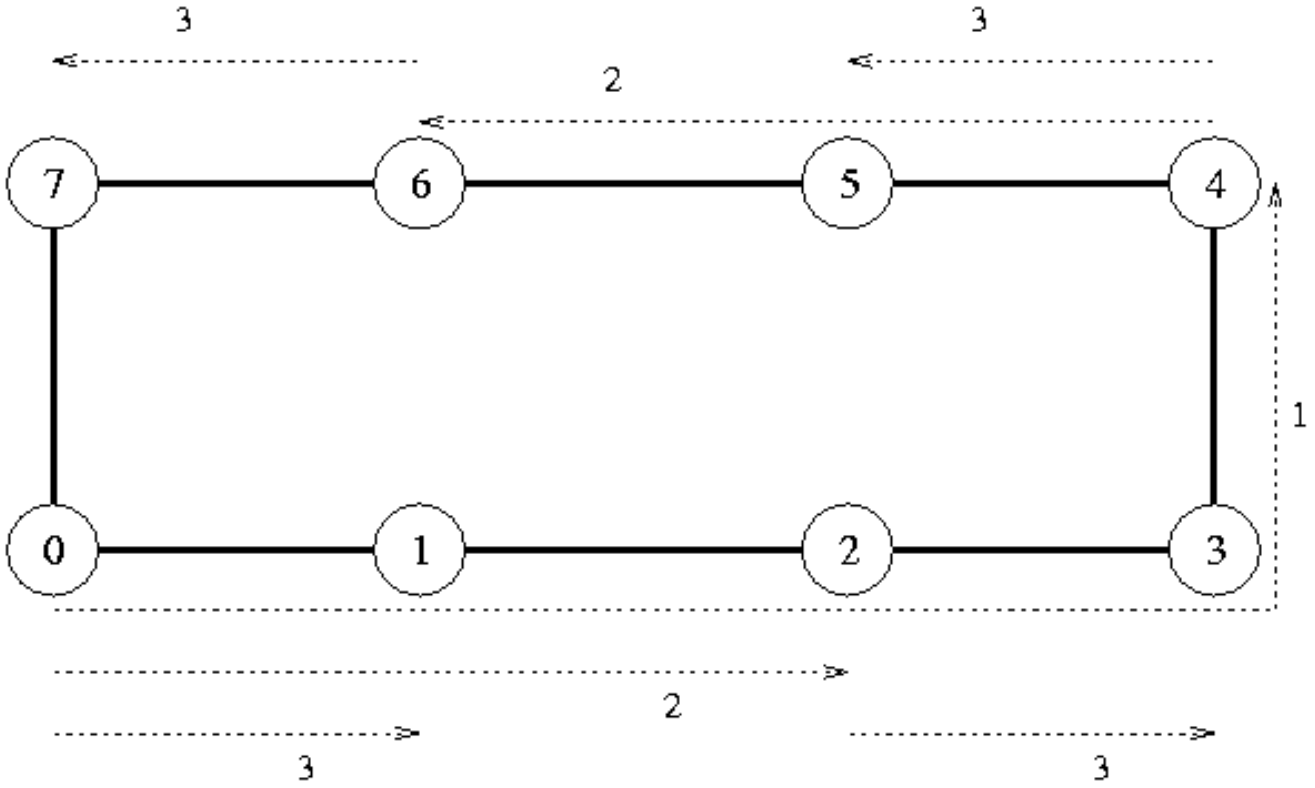
The communication time of the broadcast in case of this method is defined by the following equation:

$$t_{comm} = \sum_{i=1}^{\log_2 p} (t_s + mt_b + t_h p / 2^i) = (t_s + mt_b)\log_2 p + t_h(p-1)$$

Complexity Analysis of Communication Operations…

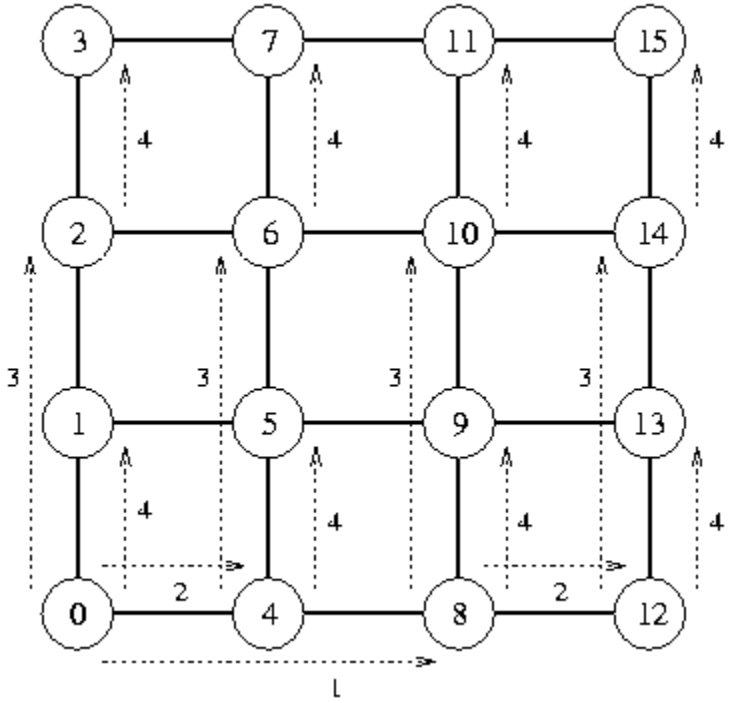
One-to-All Broadcast (*cut-through routing*)…

Ring topology



Complexity Analysis of Communication Operations…

❑ One-to-All Broadcast (*cut-through routing*)



For the **grid-torus** topology the broadcast algorithm can be obtained using the method of data communication applied to the ring network structure. The same generalization method, that is used data communication method, can be also applied.

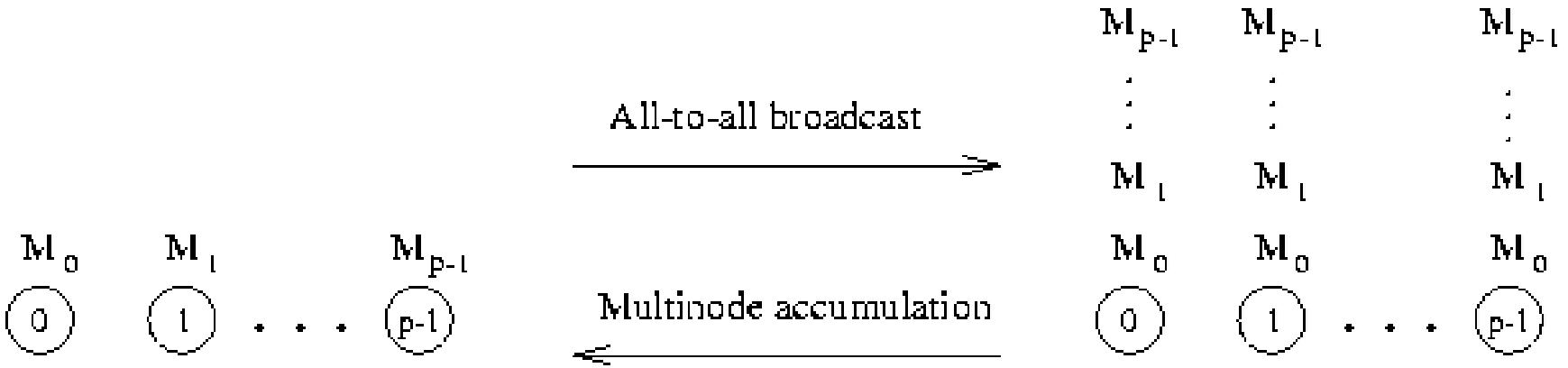
The algorithm, which is obtained, is characterized by the following relation for estimating execution time:

$$t_{comm} = (t_s + mt_b)\log_2 p + 2t_h(\sqrt{p} - 1)$$

Complexity Analysis of Communication Operations…

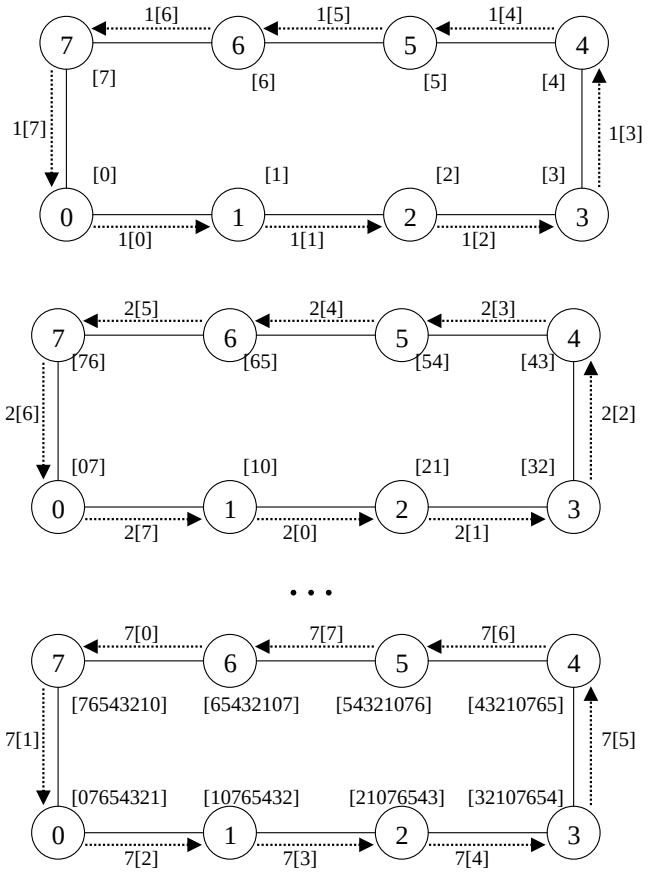
❑ All-to-All Broadcast…

All-to-all broadcast or *multinode broadcast* is a logical generalization of the single broadcast operation. *Multinode accumulation* means message receiving on every processor from all the rest network processors. So such operations are broadly used in matrix calculations.



Complexity Analysis of Communication Operations…

❑ All-to-All Broadcast (*store-and-forward routing*)…



For the **ring** topology each processor can initiate sending its message simultaneously (in any chosen direction in the ring). At any moment of time each processor receives and sends data. The multinode broadcast operation will be terminated in (p-1) data communication steps.

The duration of the broadcast execution is estimated as:

$$t_{comm} = (t_s + mt_b)(p-1)$$

Complexity Analysis of Communication Operations…

❑ All-to-All Broadcast (*store-and-forward routing*)…

For the **grid-torus** topology the multinode data broadcast can be carried out by means of the algorithm, which was obtained by generalizing the method of data communication in the ring structure:

– At *the first stage* the messages are sent separately to all the network processors located on the same horizontal lines (as a result, enlarged messages of $m\sqrt{p}$ sizes, which unite all the messages on this horizontal line, are formed on every processor of the horizontal line). The stage execution time is:

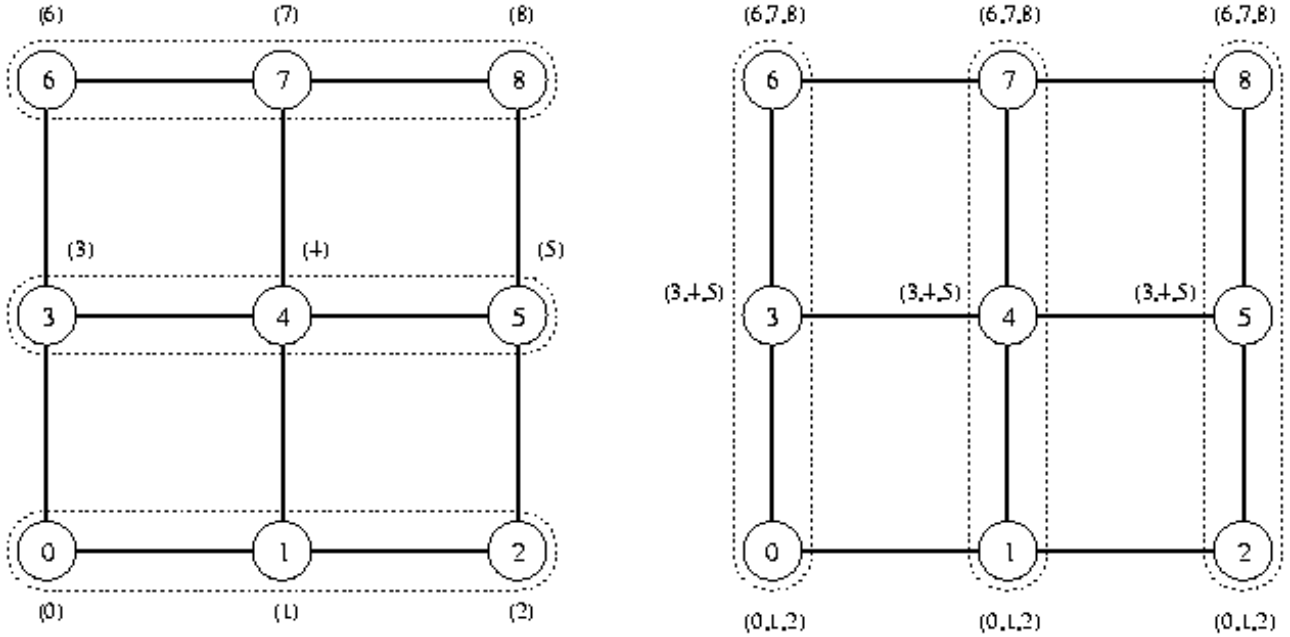
$$t_{comm}^{'} = (t_s + mt_b)(\sqrt{p} - 1)$$

– At *the second stage* the data broadcast is carried out among the processors, which form the vertical lines of the grid. The stage duration is:

$$t_{comm}^{''} = (t_s + m\sqrt{p}t_b)(\sqrt{p} - 1)$$

Complexity Analysis of Communication Operations…

❑ All-to-All Broadcast (*store-and-forward routing*)…



Grid-torus - the total duration of the broadcast operation is defined by the equation:

$$t_{comm} = 2t_s(\sqrt{p} - 1) + mt_b(p - 1).$$

Complexity Analysis of Communication Operations…

All-to-All Broadcast (*store-and-forward routing*)

The algorithm of multinode broadcast for the **hypercube** may be obtained by generalizing the previously described method of data communication for the grid topology:

- At each stage i, $1 \le i \le N$, of the algorithm execution all the network processors are engaged and exchange data with their neighbors along i dimensionality forming united messages,
- While organizing a communication between two processors, we assume that the communication channel is the channel between two equal-sized hypercubes of the smaller dimension. Every processor of the pair sends only those messages, that are intended for the processors of the neighbor hypercube,
- The broadcast execution time can be obtained by means of the following expression:

$$t_{comm} = (t_s + \frac{1}{2} mpt_b) \log_2 p$$

Complexity Analysis of Communication Operations…

- **All-to-All Broadcast (*cut-through routing*)**

  The use of the data communication method, which is efficient for the ring structure and the grid-torus topology, does not improve the execution time of the multinode broadcast. The reason for it is that the generalization of the operation execution algorithms for the single-node broadcast in case of the multinode broadcast leads to overloading of the data communication channels (i.e. it leads to emerging situations when there are several data packets waiting to be sent at the same moment in the same communication line). Channel overloading leads to delays in data communication, and decreases the advantages of the data communication method.

Complexity Analysis of Communication Operations…

- **All-to-All Broadcast (*reduction*)…**

  The reduction problem is a widely spread example of the multinode broadcast. This problem is defined in the most general way, as the procedure of processing data obtained on each processor in the course of the multinode broadcast (as an example of this problem it is possible to consider the problem of computing the sum of values, located on different processors, and transmitting the obtained sum to all the network processors).

Complexity Analysis of Communication Operations…

- **All-to-All Broadcast (*reduction*)…**

  The ways to solve the reduction problem can be the following:

  – **The direct approach** is to carry out the multinode broadcast operations and then to process the data on each processor separately,

  – **A more efficient algorithm** can be obtained if the single-node data accumulation operation is used on a separate processor, the data is processed on the processor and the obtained result is sent to all the network processors,

Complexity Analysis of Communication Operations…

- **All-to-All Broadcast (*reduction*)**
  - **The optimal way** to solve the reduction problem is to combine the multinode broadcast procedures and the data processing operations. In this case each processor performs the required data processing for the data obtained upon receiving the next message (for instance, adds the obtained value and the available on the processor partial sum). The time for solving the reduction problem by means of this algorithm is, in case when the size of the transmitted data has the single length (*m=1*) and the network topology is a hypercube, defined by the following expression:

$$t_{comm} = (t_s + t_b)\log_2 p$$

Complexity Analysis of Communication Operations…

All-to-All Broadcast

The **prefix sum problem** is another typical example of the multinode broadcast operation application:

$$S_k = \sum_{i=1}^{k} x_i, \quad 1 \le k \le p$$

The algorithm for solving the given problem can be obtained by means of specification of the general method of the multinode broadcast operation execution. In this case the processor performs the summation of the obtained values (but only in the case when the sending processor, which has sent the value, has a smaller number than the receiving processor).

Complexity Analysis of Communication Operations…

❑ One-to-All Personalized Communication…

One-to-all personalized communication or *single-node scatter* means that all the transmitted messages are different.

For this type of processor interaction a *single-node gather* of all the messages from the rest of the network processors is the communication operation, which is reverse in its action to the initial operations (the difference of this operation from the previously described one (single-node accumulation) is that the generalized gather operation does not imply any message interaction (of reduction type) in the process of data communication).

Complexity Analysis of Communication Operations…

One-to-All Personalized Communication…

The communication time of this operation is comparable to the complexity of the multinode data broadcast. The sending processor sends a message of ***m*** size to each network processor. Thus, the lower estimation of the operation execution duration may be characterized by the value $mt_b(p-1)$

Complexity Analysis of Communication Operations…

❑ One-to-All Personalized Communication (*store-and-forward routing*)

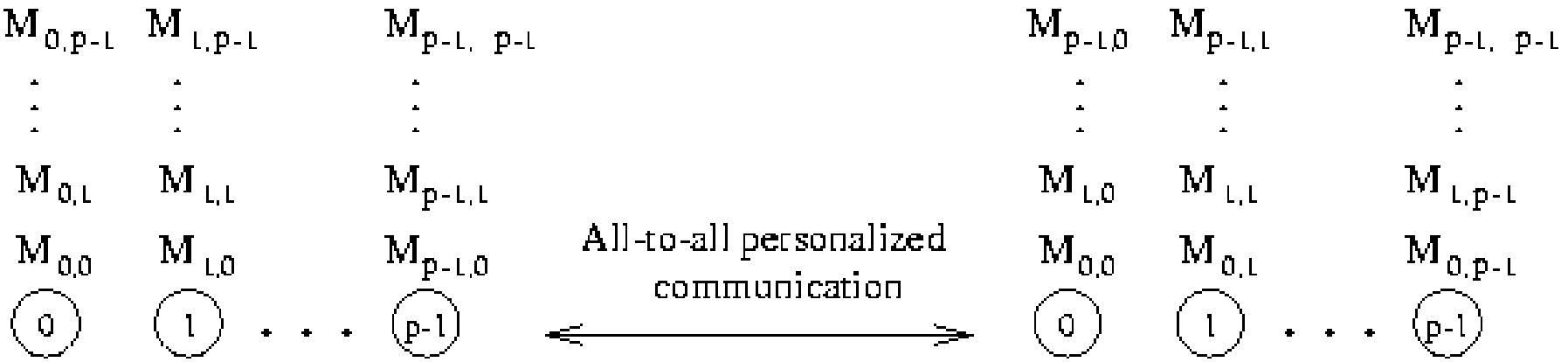
On the **hypercube** topology a possible way to execute the operation is the following: the sending processor transmits half of its messages to one of its neighbors, for instance, along the first dimension. As a result, the initial hypercube in bisected and two equal-sized hypercubes are obtained. Each of them contains half of the initial data. The further operations of message broadcast can be repeated and the total number of iterations is defined by the initial hypercube dimension. The duration of the operation can be characterized by the following equation:

$$t_{comm} = t_s \log_2 p + mt_b(p-1)$$

Complexity Analysis of Communication Operations…

Total Exchange

Total exchange is the most general case of communication interactions. The need for such operations occurs in the Fast-Fourier-Transform algorithm, matrix calculations, etc.



Complexity Analysis of Communication Operations…

- **Total Exchange (*store-and-forward routing*)…**

  Ring topology

  Each processor transmits its initial messages to its neighbor (in any chosen direction in the ring). The processors further receive the transmitted data. Then they choose their messages in the received information. After that they send the remaining part of the data further in the ring.

  The duration of such communications is estimated by means of the expression:

$$t_{comm} = (t_s + \frac{1}{2} mpt_b)(p-1)$$

Complexity Analysis of Communication Operations…

- **Total Exchange (*store-and-forward routing*)…**

  Grid-torus

  At *the first stage* data communication is arranged separately to all the processors, which are located on the same horizontal lines (only these initial messages, which should be sent to the processors of the corresponding vertical line, are sent to each processor on the horizontal line). After the termination of the first stage, there are ***p*** messages on each processor. These messages should be sent along one of the vertical lines. At *the second stage* data is sent to the processors, which form the vertical lines.

  The total duration of these operations is defined by the equation:

$$t_{comm} = (2t_s + mpt_b)(\sqrt{p} - 1)$$

Complexity Analysis of Communication Operations…

- **Total Exchange (*store-and-forward routing*)…**

  Hypercube

  At each stage i, $1 \le i \le N$, of the algorithm execution all the network processors exchange their data with their neighbors along i dimension. The communication channel in arranging the interaction between two neighbors is considered as a link between two equal-sized subhypercubes of the initial hypercube. Each processor in the pair sends to the to other processor only the messages, which are intended for the neighboring subhypercube processors.

  The broadcasting time may be obtained by means of the following equation:

$$t_{comm} = (t_s + \frac{1}{2} mpt_b) \log_2 p$$

Complexity Analysis of Communication Operations…

❑ Total Exchange (*cut-through routing routing*)

As in case of total broadcasting the application of the packet communication method does not lead to the improvement of time characteristics for the total broadcast operation.

***Hypercube*.** Broadcasting can be carried out in *p-1* sequential iterations. All the processors are split into interacting pairs of processors at each iteration. This splitting should be done in such a way that the messages transmitted among the pairs do not use the same communication paths.

As a result, the entire duration of the total broadcast may be determined by the following equation:

$$t_{comm} = (t_s + mt_b)(p-1) + \frac{1}{2} t_h p \log_2 p$$

Complexity Analysis of Communication Operations…

- **Circular Shift…**

  Permutation is a particular case of total broadcast. Permutation is the operation of redistributing the information among the network processors, during which each processor transmits a message to another network processor determined in a certain way. The concrete example of permutation is the *circular* ***q****-shift*. In this case each processor ***i, 1≤i≤N***, transmits the data to the processor ***(i+q)*** **mod** ***p***. Such a shift operation is used, for instance, in matrix computations.

Complexity Analysis of Communication Operations…

- **Circular Shift (*store-and-forward routing*)…**

  Grid-torus

  Let the processors be enumerated rowwise from *0* to *p-1*. At *the first stage* the circular shift with the step $q \bmod \sqrt{p}$ at each separate row is implemented (if the messages are transmitted through the right borders of the rows during the realization of the shift, then after ending this communication it is necessary to execute one position compensational shift up for the right grid column). At *the second stage* the circular shift up with the step $\lfloor q / \sqrt{p} \rfloor$ is realized for each grid column.

  The total duration of all broadcast operations is determined by the relation:

$$t_{comm} = (t_s + mt_b)(2\lfloor \sqrt{p}/2 \rfloor + 1)$$

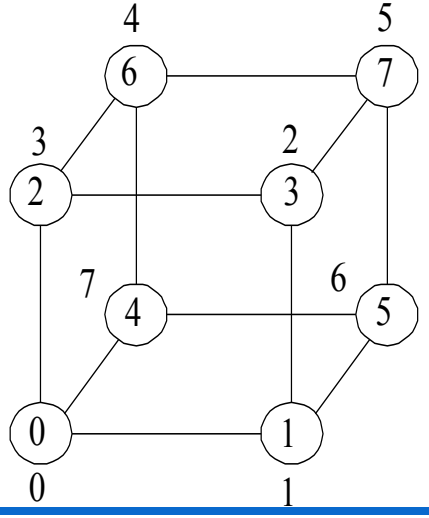
Complexity Analysis of Communication Operations…

- **Circular Shift (*store-and-forward routing*)…**

  Hypercube…

  The circular shift algorithm for the hypercube can be obtained by means of logical presentation of hypercube topology as a ring structure.

  The necessary relation can be obtained, for instance, be means of the well-known Gray code. This code can be used to determine the hypercube processors, which correspond to particular ring nodes.



Complexity Analysis of Communication Operations…

- **Circular Shift (*store-and-forward routing*)**

  Hypercube

  Let us present the value of the shift ***q*** as a binary code. The number of non-zero code positions determines the number of stages in the realization scheme of the circular shift operation.

  The shift is performed at each stage. The value of the shift step is determined by the high-order non-zero position of the value ***q*** (for instance, if the initial shift value is $q=5=101_2$, the shift with step 4 is performed at the first stage, at the second stage the step of the shift is equal to 1). The execution of each step (except the shift with step 1) consists in communication the data along the path, which includes two communication lines.

  As a result, the upper estimation for the duration of the circular shift execution is determined by the following relation:

$$t_{comm} = (t_s + mt_b)(2\log_2 p - 1)$$

Complexity Analysis of Communication Operations

Circular Shift (*cut-through routing*)

The application of packet communication can increase the efficiency of the circular shift execution for the hypercube topology.

The realization of all the necessary communication operations can be provided if each processor sends the transmitted data immediately to the receiving processors.

The use of the *dimension-ordered routing* helps to avoid collisions when communication channels are used.

The value of the longest path in this case if defined as $log_2 p$-$\gamma(p)$, where $\gamma(p)$ is the greatest integer value j , such that 2^j is the divisor of the shift value q.

The duration of the circular shift operation can be determined by means of the following expression:

$$t_{comm} = t_s + mt_b + t_h(\log_2 p - \gamma(q))$$

Logical Presentation of Network Topology…

- A number of data communication algorithms can be described in a simpler way if certain network topologies of interprocessor connections are used
- Many communication methods can be obtained by means of specific logical presentation of the given network topology

*The possibility of **logical presentation** of various topologies on the basis of concrete physical interprocessor structure is essential in parallel computations*

Logical Presentation of Network Topology…

- The methods of logical presentation (mapping) of the topologies are characterized by the three following basic characteristics:
  - ***arc congestion*** - it is expressed as the maximum number of arcs of the logical topology mapped onto a communication channel of the physical topology,
  - ***arc dilation*** - it is determined as the path of the maximum physical topology length, onto which the logical topology arc is mapped,
  - ***vertex expansion*** - it is calculated as the relation of the number of vertices in the logical and the physical topologies.

Logical Presentation of Network Topology…

- **Presentation of the Ring Topology as a Hypercube…**

  Relation between the ring topology and the hypercube can be set by means of the *binary reflected Gray code* determined in accordance with the following relations:

$$G(0,1)=0, \quad G(1,1)=1, \quad G(i,s+1)=\begin{cases} G(i,s), & i<2^s, \\ 2^s+G(2^{s+1}-1-i,s), & i\geq 2^s, \end{cases}$$

  where ***i*** gives the number of the Gray code value, and ***N*** is the code length.

Logical Presentation of Network Topology…

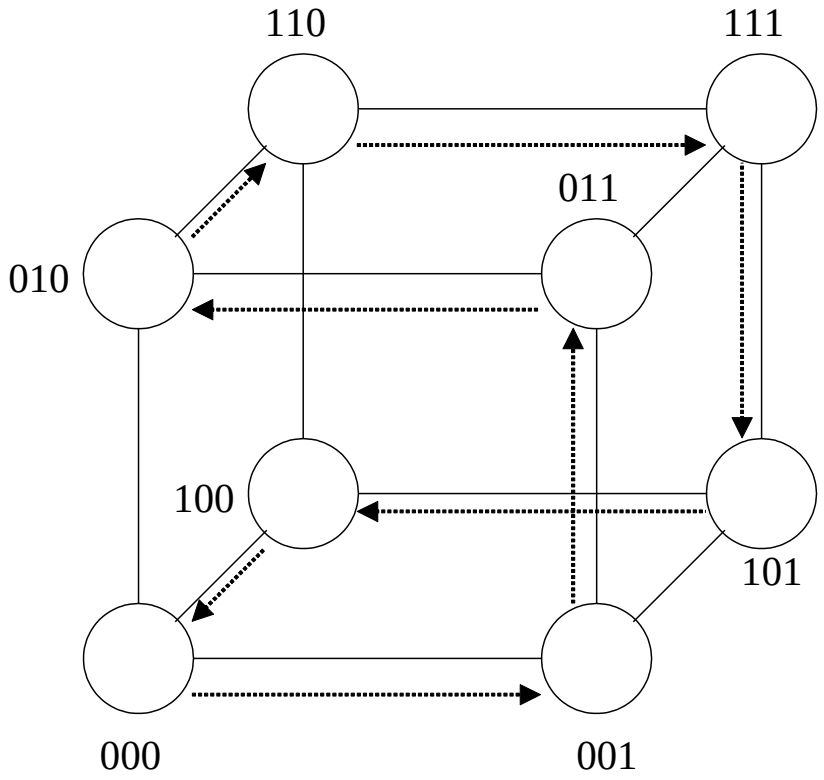
- **Presentation of the Ring Topology as a Hypercube…**

  Mapping of the ring topology onto the hypercube by means of Gray code, when the number of processors is *p=8*:

Gray code for N=1	Gray code for N=2	Gray code for N=3	Processor number	
			Hypercube	Ring
0	0 0	0 0 0	0	0
1	0 1	0 0 1	1	1
	1 1	0 1 1	3	2
	1 0	0 1 0	2	3
		1 1 0	6	4
		1 1 1	7	5
		1 0 1	5	6
		1 0 0	4	7

Logical Presentation of Network Topology…

❑ Presentation of the Ring Topology as a Hypercube



The essential characteristic of the Gray code is the fact that the neighboring values *G(i,N)* and *G(i+1,N)* differ only in a bit position. As a result, the neighboring nodes in the ring topology are mapped onto the neighboring hypercube processors.

Logical Presentation of Network Topology

❑ Mapping of the Grid Topology onto the Hypercube

The mapping of the grid topology onto the hypercube can be obtained within the frameworks of the approach, which was used for the ring structure. Then it is necessary to adopt the rule, which says that the grid element with coordinates ***(i,j)*** will correspond to the hypercube processor number

$$G(i,r)\,||\,G(j,s),$$

where the operation $||$ means the Gray code concatenation. This rule can be adopted for mapping the grid $2^r \times 2^s$ onto the hypercube of $N=r+s$ dimension.

Complexity Estimation of Communication Operations for Cluster Systems…

- One of the most efficient methods of constructing the communication environment for cluster computational systems is using *hubs* and *switches* for joining cluster processors into one computational network
- In these cases the cluster network topology is *complete graph*. There are, however, certain limitations on communication operation simultaneity:
  - Data communication at any given moment of time can be executed only between two processors, if *hubs* are used,
  - Switches can provide the interactions of several non-intersecting pairs of processors
- Another solution, which is widely used in constructing the clusters, consists in using *packet communication techniques* (which are realized, as a rule, on the basis of *TCP/IP protocol*)

Complexity Estimation of Communication Operations for Cluster Systems…

- The duration of the communication operation between two processors can be estimated according to the following expression (***model A***):

$$t_{comm}(m) = t_s + m \cdot t_b + t_h$$

- **Remarks**:
  - In this model the time of data preparation is assumed to be constant (it does not depend on the amount of the transmitted data),
  - the time of control data communication does not depend on the number of the transmitted packets, etc.

  These assumptions do not fully coincide with the practice, and the time estimations obtained by means of this model can be not accurate enough

Complexity Estimation of Communication Operations for Cluster Systems…

- ***Advanced model B***:

$$t_{comm} = \begin{cases} t_{s_0} + m \cdot t_{s_1} + (m + V_h) \cdot t_b, & n = 1 \\ t_{s_0} + (V_{\max} - V_h) \cdot t_{s_1} + (m + V_h \cdot n) \cdot t_b, & n > 1 \end{cases}$$

$n=[m/(V_{max}-V_h)]$ - is the number of packets, into which the transmitted message is partitioned,

V_{max} - defines the maximum size of the packet, which may be delivered in the network,

V_h - the volume of control data in each of the transmitted packets,

t_{s0} - in the above relations characterizes the hardware latency,

t_{S1} - defines the time for preparing a data byte for communication.

As a result, the **latency** value :

$$t_s = t_{s_0} + (V_{\max} - V_h) \cdot t_{s_1}$$

Complexity Estimation of Communication Operations for Cluster Systems…

- It is necessary to estimate the values of the parameters for the relations being used in order to apply the above described models in practice
- In this respect sometimes it can be reasonable to use simpler methods of computing the time expenses of data communication. One of the best known schemes of this type is ***the Hockney model***, which estimates the duration of the communication between two processors according to the equation (***model C***):

$$t_{comm}(m) = t_s + m t_b$$

Complexity Estimation of Communication Operations for Cluster Systems…

❑ Computational experiments

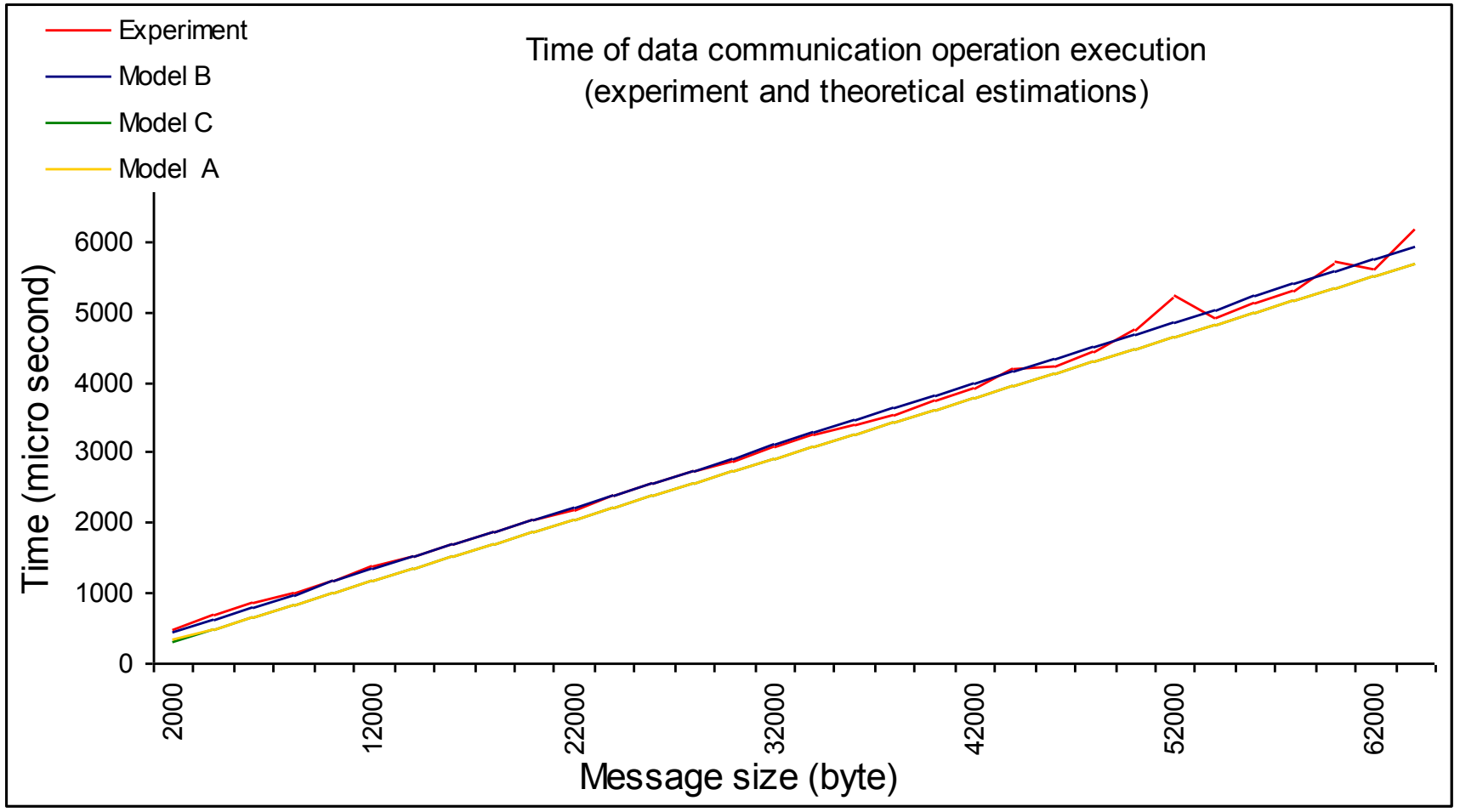
- The experiments were carried out in the network of the multiprocessor cluster of Nizhni Novgorod State University (computers IBM PC Pentium 4 1300 Mhz and Fast Etherrnet network). Communication operations were realized by means of the MPI library in these experiments,
- The latency value for the models A and C was determined as the time of transmitting the message of zero length,
- The value of network bandwidth is set to the maximum value, which was achieved in the experiments, i.e.

$$R = \max_{m} \quad (t_{comm}(m)/m)$$

  and it was assumed that $t_b = 1/R$,
- The values t_{s0} and t_{s1} were estimated by means of linear approximation of the data communication time for messages beginning with 0 size up to V_{max} size.

Complexity Estimation of Communication Operations for Cluster Systems…

- **Results of the computational experiments**



Complexity Estimation of Communication Operations for Cluster Systems…

❑ Results of the computational experiments

Message Size (byte)	Communication time (microseconds)	The Errors of the Communication Models, %		
		Model A	Model B	Model C
2000	495	33.45%	7.93%	34.80%
10000	1184	13.91%	1.70%	14.48%
20000	2055	8.44%	0.44%	8.77%
30000	2874	4.53%	-1.87%	4.76%
40000	3758	4.04%	-1.38%	4.22%
50000	4749	5.91%	1.21%	6.05%
60000	5730	6.97%	2.73%	7.09%

Complexity Estimation of Communication Operations for Cluster Systems

- The results of the experiments demonstrate that the estimations of communication time according to model B have the least error
- It should be also noted that model C accuracy can appear to be sufficient for the preliminary analysis of the time expenses of communication operations. Besides, this model is the simplest one among all the models, which have been discussed
- With regard to the latter fact we will be using model C (*the Hockney model*) in all the further sections for estimating the communication time; the recording format for the model is:

$$t_{comm}(m) = \alpha + m / \beta,$$

where α is the latency of the data communication network (i.e. $\alpha=t_s$), β is the network bandwidth (i.e. $\beta=R=1/t_b$).

Summary…

- First subsection gives the general overview of the routing algorithms and data communication methods. The method of *store-and-forward routing* and *cut-through routing* method are analyzed. The time of communication operations is estimated for these methods.
- Second subsection defines the basic types of data communication operations, which are carried out during parallel computations. Data communication for all the operations are considered for the ring, the grid and the hypercube topologies. For each of the algorithms described there are estimations of the time both for message communication and packet communication.

Summary

- The methods of logical presentations of topologies are described on the basis of physical interprocessor structures.
- Last subsection describes the models, which can help to estimate the time of data communication operations for cluster computational systems. For comparing the accuracy of different time estimations the results of the experiments are described. These results make possible to find the most precise model (model B). Besides, it is noted that for the preliminary analysis of the time communication complexity the simpler model (the Hockney model) can be more efficient.

Discussions

- Comparison of different data communication mechanisms
- Different possible basic data communication operations
- Advantages of logical topologies usage
- Is the set of basic data communication operations complete?

Exercises

- Develop the execution algorithms of the basic data communication operations for the network topology in the form of a 3-dimensional grid.
- Develop the execution algorithm of the basic data communication operations for the network topology in the form of the binary tree.
- Use model B for the estimation of the time complexity of data communication operations. Compare the obtained results.
- Use model C for the estimation of the time complexity of data communication operations. Compare the obtained results.
- Develop the algorithms of logical presentation of the binary tree for various physical network topologies.

References…

- **Gergel, V.P.** Theory and Practice of Parallel Computations. - Moscow: INTUIT.RU, 2007. - 424 p. (In Russian)
- **Gergel, V.P.** High Performance Computations for Multiprocessor Multicore Systems. - Moscow: MSU, 2010. – 544 c. (In Russian)

References

- **Andrews, G. R.** (2000). Foundations of Multithreaded, Parallel, and Distributed Programming.. – Reading, MA: Addison-Wesley
- **Culler, D., Singh, J.P., Gupta, A.** (1998) Parallel Computer Architecture: A Hardware/Software Approach. - Morgan Kaufmann.
- **Kumar V., Grama, A., Gupta, A., Karypis, G.** (1994). Introduction to Parallel Computing. - The Benjamin/Cummings Publishing Company, Inc. (2nd edn., 2003)
- **Quinn, M. J. (2004).** Parallel Programming in C with MPI and OpenMP. – New York, NY: McGraw-Hill.

Next Section

- **Parallel Programming with MPI**

Author's Team

Gergel V.P., Professor, Doctor of Science in Engineering, Course Author

Grishagin V.A., Associate Professor, Candidate of Science in Mathematics

Abrosimova O.N., Assistant Professor (chapter 10)

Kurylev A.L., Assistant Professor (learning labs 4,5)

Labutin D.Y., Assistant Professor (ParaLab system)

Sysoev A.V., Assistant Professor (chapter 1)

Gergel A.V., Post-Graduate Student (chapter 12, learning lab 6)

Labutina A.A., Post-Graduate Student (chapters 7,8,9, learning labs 1,2,3, ParaLab system)

Senin A.V., Post-Graduate Student (chapter 11, learning labs on Microsoft Compute Cluster)

Liverko S.V., Student (ParaLab system)

About the project

The purpose of the project is to develop the set of educational materials for the teaching course “Multiprocessor computational systems and parallel programming”. This course is designed for the consideration of the parallel computation problems, which are stipulated in the recommendations of IEEE-CS and ACM Computing Curricula 2001. The educational materials can be used for teaching/training specialists in the fields of informatics, computer engineering and information technologies. The curriculum consists of **the training course “Introduction to the methods of parallel programming”** and **the computer laboratory training “The methods and technologies of parallel program development”**. Such educational materials makes possible to seamlessly combine both the fundamental education in computer science and the practical training in the methods of developing the software for solving complicated time-consuming computational problems using the high performance computational systems.

The project was carried out in Nizhny Novgorod State University, the Software Department of the Computing Mathematics and Cybernetics Faculty (http://www.software.unn.ac.ru). The project was implemented with the support of Microsoft Corporation.