Annales UMCS Informatica AI 2 (2004) 331-340

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Embedding as a method to improve multichannel bus topology parameters

Mirosław Mazurek
Department of Distributed Systems, Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, W. Pola 2, 35-959 Rzeszów, Poland

Abstract

This article attempts to discuss the problems in building new topologies utilizing embedding. We propose to utilize the cartesian product to describe graphs as a right mathematical solution to adjacency matrix. We applied the cartesian product of two and sets of multiple dimensions in analysis. We proposed methodology of building logical topologies based on utilization of regular and symmetric graphs. For three particular products, we performed the analysis to measure traffic intensity in the network.

1. Introduction

Currently available computation or communication tasks utilized in parallel and distributed systems show that in the row of an accidental capacity, computing power of currently used computers is not sufficient. Constant growth of computing complexity in the computer systems brings growth of systems capacity. At the beginning this growth was accomplished by enhancement to parameters of base elements utilized in building computing structures. This showed that available alternatives, offered by base of new elements are not sufficient. Therefore, it becomes necessary to search for new directions towards improvement in transmitting parameters of currently available topologies. In the process of designing topologies, besides utilizing base topologies, meaning topologies described by particular graphs with specific abilities such as: small degree, small diameter, symmetrical or a large number of nodes, we should consider embedding as a right tool to build new topologies. To achieve this goal we could use logical operations utilized on matrix of corresponding graphs, in particular the cartesian product. These new developed original topologies are characterized by predictable parameters described in base topologies. As topologies we could use those with opposite parameters (for example bus and hypercube). Currently conducted research showed that in building new multi channel communication systems regular and symmetric topologies are of special use [1]. They insure minimal computation complexity of designing process and liveness as well large scalability assuring building the network with an unlimited number of nodes.

2. Embedding as a method to improve topology parameters

The key role in computing parallel and homologous computations for example organizing computations in the network of processors is projection of one architecture within another. The problem with simulation of one network within another is defined as a problem of embedding. Embedding of one topology within another gives new capacity of building a new communication network and also to allow to formulate a problem with finding effective representation of data structures or to allocate circuits in the distributor of VLSI.

In order to characterize operation of embedding, we have to adopt particular representation of topology. Any particular network topology usually is described in the format of graphs in which all the nodes represent computers and all edges represent network connections. Indirect graphs are usually used to model communication network. Exploiting theory of graphs to describe topology could define operation of embedding as:

Definition 1. Any given graph G_1 (V_1, E_1), known as graph "host" where V_1 is the set of its nodes, but E_1 is the set of its edges. Also let us have a graph $G_2(V_2, E_2)$ known as the graph "guest", where V_2 is the set of its nodes, and E_2 is the set of its edges. Embedding φ of graph G_1 (V_1, E_1) within graph $G_2(V_2, E_2)$ is called direct projection of nodes of "host" graph V_1 within the "guest" graph V_2 for example $\varphi = V_1 \rightarrow V_2$.

The utility of embedding results is further emphasized by the fact that many of the existing popular architectures can be modeled as product networks. An embedding of a „guest" graph G_2 in a „host" graph G_1 is a mapping of the vertices of G_1 into the vertices of G_2 and the edges of G_1 into paths in G_2. The main cost measures used in embedding efficiency are:

1) *Load* of embedding is the maximum number of nodes of G_1 mapped to any nodes G_2.
2) *Dilation* of embedding is the maximum path length in G_2 representing an edge of G_1.
3) *Congestion* of embedding is the maximum number of paths that share any edge of G_2.

If G_1 can be embedded in G_2 with load *l,* dilation *d* and congestion *c*, G_2 can emulate *t* steps of a computation running on G_1 in $O(l+d+c)t$ steps. If the values *l, d,* and *c* are constant, the slowdown introduced by this emulation is also constant.

3. Regularity and symmetry

Classical communication topologies characterize degree of its nodes determined by the number of nodes. The moment of large increase in the number of nodes usually leads to the increase in a diameter, which is a great disadvantage. Of many available topologies none of them could be considered useful to build modern computing systems because of parameters. Architecture of the parallel system should be characterized by its symmetry, which allows implementing simple rules of routing and also should not be characterized by a large number of nodes, which in result keep the cost low. However, considering them in context of particular attributes, it shows that particular systems with fixed criteria could utilize some base topologies. With regards to hypercube topologies its attention brings very good parameters, such as: small diameter, regularity, large connectivity, ability to embed simple mechanisms of routing (comes from symmetry), and fault-tolerance. Moreover, hypercube topologies give ability to imitate different structures in place of structure nodes simplifying the embedding process.

For example if we place in all nodes of cube ring of 3 nodes we get back very popular topology CCC. The disadvantage of the above architecture is limited scalability, which eliminates it from being utilized in large systems. Regularity and symmetry guarantee homogeneous topologies, which minimize cost of building and further exploitation [2]. By exploiting these kinds of topologies the process of modeling and also analysis becomes simpler and it influences improvement of parameters. Based on the theory of graphs, a graph is called regular, if every pair of nodes has the same nodes degree. A graph is called node-symmetric, if it is viewed the same from all of its vertices or from each branch. For cleaner analysis of these parameters we illustrate base definition of regularity and symmetry by utilizing the method of similarity.

Definition 2. A topology is called regular if each node has the same node degree. If any node has different node degree a topology is called irregular.

In order to describe precisely symmetry we need to reconsider the definition of similarity:

Definition 3. Two nodes u and v in a graph G are similar if for some automporphism ρ of G, $\rho(u)=v$ with $u, v \in V$. Two edges (u_1,v_1) and (u_2,v_2) in graph G are similar, if for some automporphism ρ of G, $\rho((u_1,v_1))= (u_2,v_2)$ with $(u_1,v_1), (u_2,v_2) \in E$.

Definition 4. A graph is called node-symmetric, if every pair of nodes is similar. A graph is called edge-symmetric, if every pair of edges is similar. A graph is called symmetric, if it is node-symmetric and edge-symmetric.

By analyzing symmetry we have to notice, that symmetric graph by its nodes could be symmetric by its edges, although graph which is symmetric by its edges it is not symmetric by its nodes. Definition 4 describes only which of the graphs is symmetric by its edges and which is symmetric by its nodes. But it is useless when comparing degree of symmetry particular high capacity network systems (does not allow to describe which graph is less or more symmetric), therefore it is required to introduce definition of s-symmetry [3].

Definition 5. A graph G(V,E) is said to be s-symmetric if the number of nodes can be divided into s subsets V_1,,V_2,...V_s with $\bigcup_{i=1}^{s} V_i = V$ and $\bigcap_{i=1}^{s} V_i = 0$, and in each subset V_i every pair of nodes must be similar (analogous for edges).

4. Cartesian Product as an instrument of embedding

To generate different communication topologies based on those currently existing, we exploit graphical operations. They allow building topologies characterized by incomparable parameters. If we talk about connection network in the context of logical operations, we have to notice the fact that each newly built topology could be characterized by inheritance of outgoing graphs parameters and all of its characteristics as well as by new features. They result from operation attributes, which are parameterized through character and design of the graphs. Of the primary Boolean operations such as conjunction, disjunction, symmetric difference, rejection, composition and cartesian product the last one deserves a special emphasis, because of its benefit and simplicity in operations [4].

In general a Boolean operation $G_1 x G_2$ results in a graph G with $V = V_1 x V_2$, the cartesian product of the node sets from graph G_1 and graph G_2. A node G is labeled $(u_1 \mid u_2)$ with $u_1 \in V_i$ and E of G is expressed in terms of the edges in E_1 and E_2 depending on the Boolean operation applied. Most of the Boolean operations could be described as operations on matrix of graphs G1 and G2. Therefore the cartesian product could be defined as follows:

Definition 6. A cartesian product of two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ is graph G_1xG_2, whose set of its nodes is V_1xV_2 and set of its edges is defined as follows: if $\{u_1, u_2\} \in V_1$ and $\{v_1, v_2\} \in V_2$ then (u_1v_1, u_2v_2) is edge of G_1xG_2 and only then if $u_1 = u_2$ and $(v_1,v_2) \in E_2$ or $v_1 = v_2$ and $(u_1, u_2) \in E_1$.

The example of two-dimensional cartesian product of second degree of de Bruijn's Graph and the star of five nodes is shown in Fig. 1.

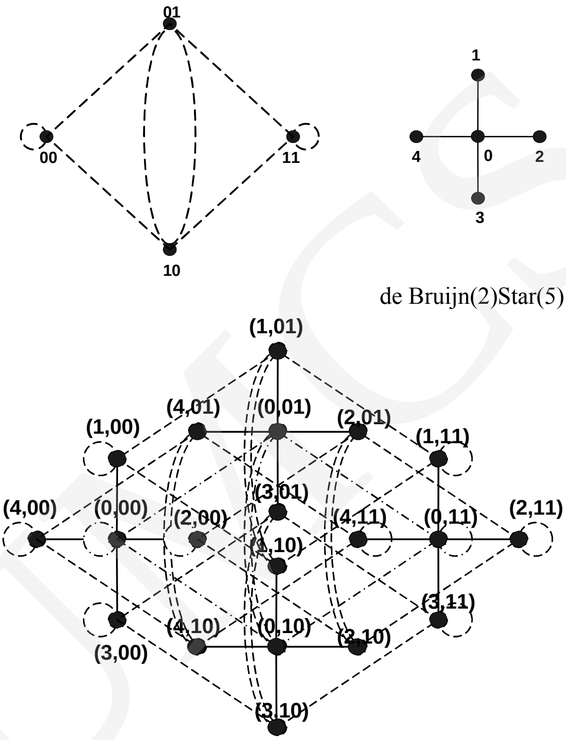

Fig. 1. Example of two-dimensional cartesian product of de Bruijn(2) and Star(5)

5. Attributes of cartesian product

Current research shows, that the best solution for connection network between all Boolean operations is the cartesian product. First of all it provides a limited number of solution nodes, secondly all ability to achieve a small diameter. It also allows connecting advantages provided by popular topologies, acquiring new topology that inherits advantages of both above topologies. It is possible to find topology X which is better in scalability than topology Y and also is better in reduction of degree or diameter but it is not better considering processing of broadcasting, for example star or hypercube. Therefore of all the above operations the cartesian product is considered the only right choice within the operations allowing building new networks. The cartesian product operation allows achieving absolute network components. The additional attribute of this operation is the fact that the number of nodes grows in a multipliable way, but diameter and average distance grow in an addable way, thus cutting costs of building network. This new structure is also characterized by simpler implementation of network algorithms, based on base and popular in topology algorithms. These attributes are specifically useful in systems scalability and allow increasing network size the same time retaining a stable number of nodes. Dependencies of the cartesian product parameters in regard to input topologies describe line of definitions which could be found in [5].

6. r-dimensional product

Besides two-dimensional operation, multi-dimensional operation is of special usage in creation of new topologies. The *r*-dimensional product of *G(N)* is denoted $PG_r(N)$, with the subscript *r* representing the number of dimensions. The *r*-dimensional product we can identify as:

Definition 7. For a given graph G(N), then its r – dimensional cartesian product labeled as $PN_r(n)$ is equal to $G_1(V_1, E_1)$, $G_2(V_2, E_2)$, ..., $G_N(V_N, E_N)$ where set of nodes $V = V_1, V_2, V_3 ... V_N = \{u_1u_2...u_n \mid u_1 \in V_1, u_2 \in V_2, ..., u_n \in V_n\}$ and set of edges $E = \{(u_1u_2...u_n, v_1v_2...v_n \mid$ where i exists where $(u_i,v_i) \in V_i$ and for each $j \neq i$ we have $u_j = v_j\}$.

Notice that the nodes in dimension *i* form copies of the graph G_i. For example when using the linear array $L_n(V,E)$ defined as $V = \{0,1,...,n-1\}$ and $E = \{(i,i+1) | i = 0,1,...,n-2\}$ and also ring $P_n(V,E)$ of *n* nodes defined as $V = \{0,1,...,n-1\}$ and $E = \{(i,i+1 \bmod n) | i = 0,1,...,n-1\}$, simply by using the product operation we could arrive at *r*-ary – n–cube, multidimensional grid and a multidimensional torus. Notice that grid a $p_1 x\ p_2 x\ ..x\ p_r$ is the product network $L_{n1}\ L_{n2}\ ...L_{nr,}$ but torus $p_1 x\ p_2 x\ ..x\ p_r$ is the product network $P_{n1}\ P_{n2}\ ...P_{nr}$. Similarly the binary *r*–dimensional cube Q_r is a product of the graph which contains one edge *Q(V={0,1}, E={(0,1)})*.

One of the fundamental parameters in communication network is the number of nodes and connections in the above network. Comparison of primary parameters of two- and multi-dimensional products is shown in Table 1.

Table 1. Comparison of primary parameters of two-and multi-dimensional products

	$G(N)$	$PG_r(N)$
Number of nodes	N	N^r
Number of edges	E	ErN^{r-1}
Diameter	d	Rd
Connectivity	κ	$r\kappa$
Min. vertex degree	δ	$R\delta$
Max. vertex degree	Δ	$r\Delta$
Partitionability	–	N^i, *dla* $i=0...r$
	k	k^r

7. Communication capabilities of product networks

In order to compare the communication capabilities of product networks a communication model is needed. The proposed model is a time-slotted packet switching model where packets are of fixed size. In each time slot, a processor can simultaneously send and receive a packet. It can also generate a single packet according to a Bernoulli distribution of parameter p (p is the packet generation rate of every node). Destination nodes of generated packets are selected uniformly randomly. Each node contains a single unlimited buffer to accommodate incoming packets. The packets are FIFO serviced. The links between processors are full duplex. If several packets are sent to the same processor, then a conflict occurs. In this case, we assume that one of the packets to be received is chosen randomly, and the others are deferred. In order to describe communication ability of any given topology we introduced concept of intensity of traffic. The *traffic intensity* of a node u in a network $G(V,E)$ is defined to be the product of the load of that node by the average service time of a packet. Since the service time is deterministic, a processor can send a packet at each time slot. Thus the average service time is assumed to be one. Therefore, the traffic intensity is reduced to the total load of the node [6].

The load of node u is the total of the locally generated load p_l (defined as the average number of packets generated by u per time unit), the load designated to this node p_d (defined as the average number of packets designated to u per time unit) and the load that has to transit throughout this node p_t (defined as the average number of packets transiting u per time unit). Therefore, the load is defined as:

$$p(u) = p_l + p_d + p_t .$$

Note that $p_l = p$, where p is the packet generation rate. Let p_e be equal to the sum of p_t and p_d. This load depends on the total number of paths used by the routing algorithm, denoted by t_u^G, that either ends at node u or have node u as an intermediate node and the traffic sent by each node to every other node. This traffic is equal to $\frac{p}{N-1}$ since each node equally sends its locally generated traffic to every other node in the system (except to itself). A node u can receive the total external load from other nodes in the system is:

$$p_e = \frac{p}{N-1} t_u^G . \tag{1}$$

The quantity $q_u^G = \frac{t_u^G}{N}\pi$ called the external load factor of node u in network G. Therefore, external load factor p_e is defined as:

$$p_e = q_u^G \cdot p \frac{N}{N-1}.$$

A node *(u_1, u_2) in G_1G_2* will be denoted $u_1\ u_2$ and the cardinality of the set of nodes V_i of a graph $G_i(V_i,E_i)$ will be denoted $|V_i|$. Consider a product network $G=G_1G_2=(V,E)$ of 2 networks $G_1(V_1,E_1)$ *and* $G_2(V_2,E_2)$, the *total external load* is defined by lemma 1.

Lemma 1. Let u_1 and u_2 be two nodes in G_1 and G_2 respectively, and $q_{u_1}^G$ and $q_{u_2}^G$ their corresponding external load factor. Then the external load factor of node u_1u_2 in $G=G_1G_2$ is:

$$q_{u_1u_2}^{G_1G_2} = q_{u_1}^{G_1} q_{u_2}^{G_2}.$$

Proof. From formula (1), the external load factor of u_1u_2 in G is:

$$q_{u_1u_2}^{G_1G_2} = \frac{t_{u_1u_2}^{G_1G_2}}{N}.$$

Where $N = |V_1| \cdot |V_2|$, i.e, the number of nodes in G_1G_2 and $t_{u_1u_2}^{G_1G_2}$ is the total number of paths that either end at u_1u_2 or have node u_1u_2 as an intermediate node. Furthermore, it can be easily concluded that $t_{u_1u_2}^{G_1G_2}$ can be expressed as $t_{u_1u_2}^{G_1G_2} = t_{u_1}^{G_1} |V_2| + t_{u_2}^{G_2} |V_1|$ since we have $|V_2|$ copies of G_1 and $|V_1|$ copies of G_2. Therefore,

$$\frac{t_{u_1u_2}^{G_1G_2}}{|V_1| \cdot |V_2|} = \frac{t_{u_1}^{G_1} |V_2| + t_{u_2}^{G_2} |V_1|}{|V_1| \cdot |V_2|} = \frac{t_{u_1}^{G_1}}{|V_1|} + \frac{t_{u_2}^{G_2}}{|V_2|} = q_{u_1}^{G_1} + q_{u_2}^{G_2}.$$

A corollary of this lemma for the product of r graphs is that:

$$q_{u_1u_n\ldots u_r}^{G_1G_2\ldots G_r} = \sum_{i=1}^{r} q_{u_i}^{G_i}.$$

Theorem 1. The traffic intensity of a node u_1u_2 in a network G_1G_2 is defined as:

$$p_{u_1u_2} = p\left(1 + \frac{N}{N-1}\left(q_{u_1}^{G_1} + q_{u_2}^{G_2}\right)\right).$$

Proof. The proof follows from formula 1 and lemma 1.

To compute the traffic intensity of multidimensional torus P_p, product networks r-ary n-cube Q_n and multidimensional mesh L_p we first need to determine the external load factor of their building blocks. Note that the hypercube network can be obtained from ${}^{r}Q_n$ by setting r=2. When $r=2$ we have K_2.

Theorem 2. The external load factor of a node u in each building block respectively for P_n, K_r, L_n is defined as:

$$q_u^{P_n} = \frac{1}{p}\left[\frac{p(p+1)}{2} - \left\lfloor \frac{p}{2} \right\rfloor^2 - \left\lceil \frac{p}{2} \right\rceil\right],$$

$$q_u^{K_r} = \frac{r-1}{r},$$

$$q_u^{L_n} = \frac{1}{p}\left[(p-u)(2u+1)-(u+1)\right].$$

The traffic intensity of each building block can be concluded from the theorem 1 and theorem 2. The traffic intensity of three showed networks, is computed in the theorem 3.

Theorem 3. The traffic intensity of a multidimensional torus $P_{n_1}P_{n_2}...P_{n_r}$ of $N = p_1 \times p_2 \times ... \times p_r$ nodes, an r-ary n-cube ${}^{r}Q_n = K_n^r$ of $N = n^r$ nodes and a multidimensional mesh $L_1 \times L_2 \times ... \times L_r$ of $N = p_1 \times p_2 \times ... \times p_r$ nodes are:

$$q_{u_1...u_r} = p\left(1+\frac{N}{N-1}\left(\sum_{i=1}^{r}\frac{1}{p_i}\times\left[\frac{p_i(p_i+1)}{2} - \left\lfloor\frac{p_i}{2}\right\rfloor - \left\lceil\frac{p_i}{2}\right\rceil\right]\right)\right),$$

$$q_{u_1...u_r} = p\left(1+\frac{N}{N-1}\left(r\times\frac{n-1}{n}\right)\right),$$

$$q_{u_1...u_r} = p\left(1+\frac{N}{N-1}\left(\sum_{i=1}^{r}\frac{1}{p_i}\times\left[(p_i-u_i)(2u_i+1)-(u_i+1)\right]\right)\right).$$

Proof. The proof follows from theorem 1 and theorem 2.

From the theorem above, we note that the *r*-ary *n*-cube and torus are symmetric and in this case the traffic intensity does not dependent on the node. In the case of a multidimensional mesh the traffic depends on the node. The traffic intensity of a hypercube can be obtained from that of a *n*-ary *r*-cube by setting $n=2$. In the case of a mesh and a torus, we consider a two-dimensional network of size xy where x and y are both even. Since the mesh is not symmetric, we take the traffic intensity in the node $u_1=y/2$ and $u_2=x/2$. The traffic intensity of a torus of $N=xy$, a hypercube of N nodes and a mesh of $N=xy$ nodes is:

$$q_{u_1 \dots u_r} = p\left(1 + \frac{N}{4(N-1)} \times \left(\frac{N}{x} + x\right)\right),$$

$$q_{u_1 \dots u_r} = p\left(1 + \frac{N \log_2 N}{2(N-1)}\right),$$

$$q_{u_1 \dots u_r} = p\left(1 + \frac{N}{2(N-1)} \times \left(\frac{N-1}{x} + x \times \left(1 - \frac{2}{N}\right)\right)\right).$$

8. Conclusions

By utilizing methods of embedding in design of new topologies we allow to accomplish new original topologies. These new structures are characterized by predictable parameters described above by attributes of the base topologies. Operation of the cartesian product by one means guarantee of a small degree of computation nodes, by other means it is possible to achieve a small diameter. Embedding allows connection of two popular topologies with fixed attributes and attaining new topology, which inherit all the above topology abilities. Additional attribute of this operation is the fact that a number of nodes grows by means of multiplicity and diameter grows by addict ability. The grown structure is also characterized by simpler structure of implementation of the network algorithms based on fundamental and commonly known algorithms in base topologies. All these attributes are particularly useful in systems scalability and allow to increase network size although, retaining stabilized number of nodes.

References

[1] Hajder M., Dymora P., Mazurek M., *Improvement methods of multichanel bus topologies parameters*, Sixth International Conference Soft Computing and Distributed Processing – WSIiZ Rzeszów, (2001).

[2] Hajder M., Mazurek M., *Grafy regularne w projektowaniu wielokanałowych optycznych systemów komunikacji*, Zeszyty Naukowe Politechniki Rzeszowskiej, Informatyka, Rzeszów 1 (2003), in Polish.

[3] Kotsis G., *Interconnection Topologies and Routing for Parallel Processing Systems*, Technical Report Series, ACPC/TR 92-19, (1992).

[4] Ohring S. R., Ibel M., Das S. K., *Reliable Broadcasting in Product Networks in the Presence of Faulty Nodes*, Proceedings of the 7th Symposium on Parallel and Distributed Processing, (1995).

[5] Youssef A., *Design and Analysis of Product Networks*, Proceedings of the 5th Symposium on the Frontiers of Massively Parallel Computation, (1995).

[6] Bellachaachia A., Youssef A., *A Unified Theory for a Traffic Analysis in Product Networks*, Proceedings of the 9th International Processing Symposium, (1995).