

Classification-based Virtual Network Embedding Method and System

Technical Field

The present invention relates to network virtualization technology, and particularly relates to a classification-based virtual network embedding method and system.

Background Art

Network virtualization technology is a core internet technology in future, which allows supporting a plurality of virtual networks on a shared bottom resource. These virtual networks may either have different network frameworks or carry different types of services therebetween.

Virtual network embedding is one key problem of the network virtualization technology. Node and link demands required by the virtual network are satisfied by mapping virtual nodes and virtual links in a virtual network request on the shared bottom resource. Virtual network embedding is proven to be a technical problem which is very difficult to solve and has higher computation complexity. No effective measures are proposed in the present methods to reduce the computation complexity, so that the mapping processing time is relatively long and the utilization ratio of the shared bottom resource is lower.

Summary of the Invention

The object of the present invention is to provide a classification-based virtual network embedding method and system for overcoming the above problems.

In order to achieve the above objectives, the present invention provides a classification-based virtual network embedding method, the method comprises:

step 101) when receiving a virtual network request, classification virtual nodes related to a virtual network request according to the classification standard, and dividing the virtual nodes belonging to different classifications to different virtual network subrequests;

wherein the classification standard is namely to classify the virtual nodes into a

substrate/physical network of an area nearest thereof according to the demand information of the virtual nodes on geographical positions; and all the virtual nodes in the same virtual network subrequest belong to the same classification, and the classification is namely a classification to which any virtual node in the virtual network subrequest belongs;

step 102) sending each virtual network subrequest to a certain area in the substrate/physical network to which the virtual network subrequest belongs to; and

step 103) mapping the virtual network subrequests onto the bottom physical layer network in the area according to a load balancing demand of each node in the virtual network subrequest and the load balancing capacity of each node in the area; wherein,

the load balancing demand represents the load demand of each virtual node in case of balanced loads; and the load demand is defined as a product of resources of the virtual node request and resources of all link requests connected to the virtual node; and

the load balancing capacity represents the load capacity of each physical node in the area in case of balanced load; and the load capacity is defined as a product of available resources of the physical node and available resources of all links connected to the physical node.

When classification analysis is performed on virtual nodes in a virtual network request, the virtual nodes are classified into a substrate/physical network of an area nearest thereof according to the demand information of the virtual nodes on geographical position.

The step 103) further comprises:

step 103-1) detecting a virtual node n^v which is unmapped currently and has maximum load balancing demand in the virtual network subrequest;

step 103-2) seeking k candidate bottom physical nodes of the virtual node n^v in the area, wherein the k candidate nodes are the first k bottom physical nodes with maximum load balancing capacity which are unmapped in the area currently and can satisfy the demand of the virtual node n^v ; and

step 103-3) for each candidate bottom physical node, calculating the sum of the path length between the candidate bottom physical node and the bottom physical nodes corresponding to each mapped virtual node, mapping the virtual node n^v on the candidate node having shortest path length, and mapping the virtual links between the mapped virtual nodes.

The load balancing capacity is obtained as follows:

step 201) respectively performing normalization processing on resources of the physical nodes and physical links of each area inclusive in the substrate/physical network;

wherein, the normalization processing of the physical nodes is related to the storage capacity of the nodes, magnetic disk I/O and quantity of CPU, and memory size; and the normalization processing of the physical links is related to the band width, delay and delay jitter of the links;

step 202) calculating the load capacity of the nodes in each area in the substrate/physical network;

wherein, the load capacity is defined as a product of available resources of the physical node and available resources of all links connected to the physical node.

step 203) calculating load balancing matrix of each area in the substrate/physical network;

wherein, the elements of the balanced load matrix are: a weighted value mutually contributed by the load balancing capacities between the physical nodes in a certain area, and the weighted value mutually contributed by the load balancing capacities is particularly as follows: when the physical node n^s is just the physical node, i.e., $n^s = m^s$, or the physical node n^s and the physical node m^s are connected by a physical link, the physical node n^s contributes to the load balancing capacity of the physical node m^s , and the weighted value is a nonzero positive number at the moment; otherwise, the weight contributed by the physical node n^s to the load balancing

capacity of the physical node m^s is 0; wherein, n^s represents the n^s line of the matrix, m^s represents the m^s row of the matrix, both the line number and the row number of the matrix are the total number of the physical nodes, and a corner mark s represents the substrate/physical network; and

step 204) iteratively calculating the load balancing capacity of each physical node in the area according to the load balancing matrix, i.e., the load balancing capacity of the physical node n^s is the sum of the load balancing capacities contributed by all the physical nodes in the area.

The load balancing demand is obtained by employing the following method:

step 301) respectively performing normalization processing on the virtual node and virtual link demands in each virtual network subrequest;

wherein, the normalization processing of the virtual nodes is related to the demands on the storage capacity of the nodes, quantity of magnetic disk I/O and CPU, and memory size; and the normalization processing of the virtual links is related to the demands on the band width, delay and delay jitter of the links;

step 302) calculating the load demand of each virtual node in each virtual network subrequest;

wherein, the load demand is defined as a product of demands of the virtual node and demands of all links connected to the virtual node;

step 303) calculating the load balancing matrix of each virtual network subrequest;

wherein, the elements of the load balancing matrixes are: a weighted value mutually affected by the load balancing demands between the virtual nodes in the virtual network subrequest, and the weighted value is particularly as follows: when the virtual node n^v is just the virtual node m^v , i.e., $n^v = m^v$, or the virtual node n^v and the virtual node m^v are connected by a virtual link, the weighted value is defined to be a nonzero positive number; otherwise, the weight affected by n^v to the load balancing demand of m^v is 0;

wherein, n^v represents the n^v line of the matrix, m^v represents the m^v row of the matrix, both the line number and the row number of the matrix are the total number of the virtual nodes, and a corner mark v represents the virtual network request; and

step 304) iteratively calculating the load balancing demand of each virtual node in the area according to the load balancing matrix, i.e., the load balancing demand of the virtual node n^v being the sum of the load balancing demands affected by all the virtual nodes in the virtual network subrequest.

In order to achieve the above objectives, the present invention provides a classification-based virtual network embedding system, the system comprises:

a global management module configured to, when receiving a virtual network request, classify virtual nodes related to the virtual network request according to a classification standard, and divide the virtual nodes belonging to different classifications to different virtual network subrequests, wherein the classification standard means that one virtual node classifies the virtual nodes into a substrate/physical network of an area nearest thereof; and send each virtual network subrequest to a certain area in the substrate/physical network to which the virtual network subrequest belongs to; and

a local management module configured to map the virtual network subrequests onto the bottom physical layer network in the area according to a load balancing demand of each node in the virtual network subrequest and the load balancing capacity of each node in the area;

wherein, the load balancing demand represents the load demand of each virtual node in case of balanced loads; and the load demand is defined as a product of resources of the virtual node request and resources of all link requests connected to the virtual node;

the load balancing capacity represents the load capacity of each physical node in the area in case of balanced load; and the load capacity is defined as a product of available resources of the physical node and available resources of all links connected to the physical node.

When classification analysis is performed on virtual nodes in a virtual network

request, the virtual nodes are classified into a substrate/physical network of an area nearest thereof according to the demand information of the virtual nodes on geographical position.

The local management module further comprises:

a load balancing capacity or load balancing demand acquisition submodule configured to acquire the load balancing capacity value of each node in each area of the substrate/physical network, and acquire the load balancing demand value of each virtual node of each virtual network subrequest;

a detection submodule configured to detect a virtual node n^v which is unmapped currently and has maximum load balancing demand in the virtual network subrequest;

a candidate bottom physical node selection submodule configured to seek k candidate bottom physical nodes of the virtual node n^v in the area, wherein the k candidate nodes are the first k bottom physical nodes with maximum load balancing capacity which are unmapped in the area currently and can satisfy the demand of the virtual node n^v ; and

a node and link mapping submodule configured to, for each candidate bottom physical node, calculate the sum of the path length between the candidate bottom physical node and the bottom physical nodes corresponding to each mapped virtual node, map the virtual node n^v on the candidate node having shortest path length, and map the virtual links between the mapped virtual nodes.

The load balancing capacity acquisition submodule further comprises:

a first normalization processing module configured to respectively perform normalization processing on resources of the physical nodes and physical links of each area inclusive in the substrate/physical network;

wherein, the normalization processing of the physical nodes is related to the storage capacity of the nodes, magnetic disk I/O and quantity of CPU, and memory size; and the normalization processing of the physical links is related to the band width, delay and delay jitter of the links;

a first node load capacity acquisition module configured to calculate the load capacity of each node in the substrate/physical network;

wherein, the load capacity is defined as a product of available resources of the physical node and available resources of all links connected to the physical node;

a load balancing matrix acquisition module configured to calculate the load balancing matrix of each node in the substrate/physical network;

wherein, the load balancing matrix defines a weighted value mutually contributed by the load balancing capacities between the physical nodes in the area of the substrate/physical network, when and only when the physical node n^s is just the physical node, i.e., $n^s = m^s$, or the physical node n^s and the physical node m^s are connected by a physical link, the weighted value is a nonzero positive number at the moment; otherwise, the weight contributed by the physical node n^s to the load balancing capacity of the physical node m^s is 0;

a load balancing capacity calculation module configured to iteratively calculate the load balancing capacity of each physical node in the area according to the load balancing matrix, i.e., the load balancing capacity of the physical node n^s is the sum of the load balancing capacities contributed by all the physical nodes in the area.

The load balancing demand acquisition submodule further comprises:

a second normalization processing module configured to respectively perform normalization processing on the virtual node and virtual link demands in each virtual network subrequest;

wherein, the normalization processing of the virtual nodes is related to the demands on the storage capacity of the nodes, quantity of magnetic disk I/O and CPU, and memory size; and the normalization processing of the virtual links is related to the demands on the band width, delay and delay jitter of the links;

a second load demand calculation module configured to calculate the load demand of each virtual node in each virtual network subrequest;

wherein, the load demand is defined as a product of demands of the virtual node

and demands of all links connected to the virtual node;

a second load balancing matrix calculation module configured to calculate the load balancing matrix of each virtual network subrequest;

wherein, the load balancing matrix defines a weighted value mutually affected by the load balancing demands between the virtual nodes in the virtual network subrequest, when and only when the virtual node n^v is just the virtual node m^v , i.e., $n^v = m^v$, or the virtual node n^v and the virtual node m^v are connected by a virtual link, the weighted value is a nonzero positive number at the moment; otherwise, the weight affected by the virtual node n^v to the load balancing demand of the virtual node m^v is 0; and

a load balancing demand acquisition module configured to iteratively calculate the load balancing demand of each virtual node in the area according to the load balancing matrix, i.e., the load balancing demand of the virtual node n^v being the sum of the load balancing demands affected by all the virtual nodes in the virtual network subrequest.

In conclusion, the present invention provides a classification-based virtual network embedding method, including: performing classification analysis on the virtual nodes in the virtual network request according to the network topology information of each area, dividing the virtual network of the request into small virtual network subrequests according to the classification results, and allocating the virtual network subrequests to each local manager to perform parallel mapping; when the virtual network subrequests are being mapped by each local manager, a resource balancing mechanism is introduced to respectively perform balancing processing on the demand limit of the virtual network subrequests and the available resources of the substrate/physical network in the area, and coordinate to consolidate the mapping of the virtual nodes and virtual links at the same time; after the management entities of all the areas finish mapping, a global manager consolidates the mapping results of each local manager and feeds the results back to users. According to the virtual network embedding method, the mapping of the

virtual network request is processed on the global manager in a distributed manner, and the mapping of the virtual network requests is processed on the local manager in a centralized manner. The classification mainly utilizes such demand limit information like the geographical position of each virtual node in the virtual network request to calculate the distance between each virtual node and the management entity of each area, and perform classification analysis according to the distance so as to determine the area to which the virtual node belongs. The dividing is to divide the virtual nodes belonging to the same classification to same virtual network subrequests, wherein the virtual links between the virtual network subrequests is simplified into one virtual link. According to the virtual network embedding method, when the virtual network subrequests are being mapped by each local manager, the resource balancing mechanism is introduced. According to the virtual network embedding method, when the virtual network subrequests are being mapped by each local manager, the mapping of the virtual nodes and the mapping of the virtual links are coordinated and consolidated. The resource balancing mechanism is implemented through consolidating the self-provided resource load of the node and the effects of resource load of the neighbor node thereof on the node. The load capacity of the node is related to the resources of the node and the resources of the links connected to the node.

Compared with the prior art, the present invention has the technical advantages that:

Through the foregoing technical solution, the virtual network request will be divided into small subrequests, and concurrently mapped by each local manager, thus reducing the computation complexity and reducing the mapping time. When the virtual network subrequests are being mapped by each local manager, the resource balancing mechanism is introduced, so as to implement the load balancing of the shared bottom resource. The coordination and integration of the mapping of the virtual nodes and the mapping of the virtual links avoid wide distribution of mapping of the nodes, and reduce the bottom physical path length of mapping of the links.

Brief Description of the Drawings

Fig. 1 is a working flow chart of a global manager in a classification-based virtual network embedding method provided by the invention;

Fig. 2 is a working flow chart of a local manager in the classification-based virtual network embedding method provided by the invention;

Fig. 3 is a block diagram of a substrate/physical network framework in an embodiment of the classification-based virtual network embedding method provided by the invention;

Fig. 4 is a block diagram of virtual network embedding of the global manager in the embodiment of the classification-based virtual network embedding method provided by the invention;

Fig. 5 is a block diagram of a load balancing mechanism in the embodiment of the classification-based virtual network embedding method provided by the invention; numbers in small boxes near each node in the figure represent normalized node resources which are affected by storage capacity, magnetic disk I/O, CPU, memory size and the like; and numbers on lines in the figure represent normalized link resources which are affected by band width, delay and jitter;

Fig. 6-1 is a block diagram of a virtual network subrequest provided by the embodiment of the invention;

Fig. 6-2 is a block diagram of a substrate/physical network in an area before mapping provided by the embodiment of the invention; and

Fig. 6-3 is a block diagram of the substrate/physical network in the area after mapping provided by the embodiment of the invention.

Detailed Description of the Embodiments

The present invention will be described in details hereinafter with reference to the drawings.

A classification-based virtual network embedding method of the present invention includes:

performing classification analysis on virtual nodes in a virtual network request according to network topology information of each area, dividing the virtual network of

the request into small virtual network subrequests according to the classification results, and allocating the virtual network subrequests to each local manager to perform parallel mapping; when the virtual network subrequests are being mapped by each local manager, a resource balancing mechanism is introduced to respectively perform balancing processing on the demand limit of the virtual network subrequests and the available resources of the substrate/physical network in the area, and coordinate to consolidate the mapping of the virtual nodes and virtual links at the same time; after the management entities of all the areas finish mapping, a global manager consolidates the mapping results of each local manager and feeds the results back to users.

In order to state the contents of the present invention clearly, the mapping of the virtual network is described as follows in the present invention.

A. Substrate/physical network

The substrate/physical network is defined as an undirected graph set $G^s(N^s, E^s)$, the i area of the substrate/physical network is represented as an undirected graph $G_i^s(N_i^s, E_i^s)$, wherein N_i^s represents a node set in the i area, and E_i^s represents a link set in the area i . The neighbor node set of node n^s is defined as $N^s(n^s)$, and a link set connected to the node n^s is defined as $E^s(n^s)$. Both the nodes and links in the substrate/physical network are related to the resource limits thereof. We define that the node resource limit information in the substrate/physical network include geographical position L^s , storage capacity S^s , magnetic disk I/O rate I^s , CPU capacity C^s and memory size M^s , and the link resource limit includes band width B^s , delay D^s and delay jitter J^s (for example, the node resource limit information of the physical node n^s): the geographical position represents position coordinate $L^s(n^s) = (10, 20)$, storage capacity $S^s(n^s) = 500MB$, magnetic disk I/O rate $I^s(n^s) = 100bps$, CPU capacity $C^s(n^s) = 4$ and memory size $M^s(n^s) = 2GB$, and the link resource limit represents:

band width $B^s(n^s) = 100MB$, delay $D^s(n^s) = 200ms$ and delay jitter $J^s(n^s) = 20ms$).

The resource limit of the substrate/physical network in the present invention are listed in details in table 1, and weight affecting factors are defined for each resource limit, so as to calculate the normalization resource limit of the nodes and links. Therefore, the

normalization resource of the node n^s in the substrate/physical network is represented

as: $R_N^s(n^s) = S^s(n^s) * w_s^n + I^s(n^s) * w_I^n + C^s(n^s) * w_C^n + M^s(n^s) * w_M^n$;and the

normalization resource of the link e^s is represented as:

$R_E^s(e^s) = B^s(e^s) * w_B^e + D^s(e^s) * w_D^e + J^s(e^s) * w_J^e$, .wherein the geographical position is

mainly used for classification analysis, and does not need to be considered into the normalization represent of the node resources.

Table 1 Resource limit of substrate/physical network

Node resource limit			Link resource limit		
Factor	Symbol representation	Weig hting factor	Factor	Symbol representation	Weig hting factor
Geograp hical position	L^s		Band width	B^s	w_B^e
Storage capacity	S^s	w_s^n	Delay	D^s	w_D^e
Magnetic disk I/O	I^s	w_I^n	Delay jitter	J^s	w_J^e
Quantity of CPU	C^s	w_C^n			
Memory size	M^s	w_M^n			
Normaliz ation	R_N^s		Normaliz ation	R_E^s	

resources			resources		
$w_s^n + w_l^n + w_c^n + w_m^n = 1$			$w_b^e + w_d^e + w_j^e = 1$		

B. Virtual network request

The virtual network request is defined as an undirected graph $G^v(N^v, E^v)$, the i subrequest of the virtual network request is represented as an undirected graph $G_i^v(N_i^v, E_i^v)$, wherein N_i^v represents a node set of the i virtual network subrequest, and E_i^v represents a link set of the i virtual network subrequest. The neighbor node set of node n^v is defined as $N^v(n^v)$, and a link set connected to the node n^v is defined as $E^v(n^v)$. Both the nodes and links in the virtual network request are related to the demand limits thereof. We define that the demand limit information proposed by the virtual nodes in the virtual network request include geographical position L^v , storage capacity S^v , magnetic disk I/O rate I^v , quantity of CPU C^v and memory size M^v , and the virtual link demand limit includes band width B^v , delay D^v and delay jitter J^v (for example, the node resource limit information of the physical node n^v): the geographical position represents position coordinate $L^v(n^v) = (30, 20)$, storage capacity $S^v(n^v) = 5MB$, magnetic disk I/O rate $I^v(n^v) = 50bps$, CPU capacity $C^v(n^v) = 2$ and memory size $M^v(n^v) = 200MB$, and the link resource limit represents: band width $B^v(n^v) = 10MB$, delay $D^v(n^v) = 200ms$ and delay jitter $J^v(n^v) = 20ms$). The demand limit of the virtual network request in the present invention are listed in details in table 2, wherein the weight affecting factors of each demand limit are identical to the weight affecting factors of the resource limit in the substrate/physical network. Therefore, the normalization resource of the virtual node in the virtual network request is represented as: $R_N^v(n^v) = S^v(n^v) * w_s^n + I^v(n^v) * w_l^n + C^v(n^v) * w_c^n + M^v(n^v) * w_m^n$; the

normalization resource of the virtual link e^v is represented as:

$R_E^v(e^v) = B^v(e^v) * w_B^e + D^v(e^v) * w_D^e + J^v(e^v) * w_J^e$, wherein the geographical position is mainly used for classification analysis, and does not need to be considered into the normalization represent of the node resource demands.

Table 2 Resource limit of virtual network request

Virtual node demand limit			Virtual link demand limit		
Factor	Symbol representation	Weighting factor	Factor	Symbol representation	Weighting factor
	L^v		Band width	B^v	
Storage capacity	S^v	w_S^n	Delay	D^v	w_D^e
Magnetic disk I/O	I^v	w_I^n	Delay jitter	J^v	w_J^e
Quantity of CPU	C^v	w_C^n			
Memory size	M^v	w_M^n			
Normalization resources	R_N^v		Normalization resources	R_E^v	
$w_S^n + w_I^n + w_C^n + w_M^n = 1$			$w_B^e + w_D^e + w_J^e = 1$		

C. Position center of substrate/physical network in an area

In order to better map the virtual network request into the substrate/physical network, the topology information of the virtual network request is sufficiently utilized in the present invention, and classification analysis is performed on the virtual nodes in

the virtual network request according to the demand limit of the virtual nodes on geographical positions. The position centers of each area in the substrate/physical network are defined as O^s , wherein O_i^s is the position center of area i , which is defined as the geometric center of all the nodes in the area i , i.e.:

$$O_i^s = \frac{1}{|N_i^s|} \sum_{\forall n^s \in N_i^s} L^s(n^s)$$

Wherein, $|N_i^s|$ represents the number of nodes in the area i .

D. Load balancing mechanism

In order to improve the resource utilization ratio of the substrate/physical network and load balancing, a load balancing mechanism is employed in the present invention to sufficiently utilize the self-provided resources of the node and resources of neighbor nodes. In the substrate/physical network, the load capacity of the node is related to the resources of the node and the resources of the links connected to the node, and the resources of the links are shared by the two ends of the node. Therefore, the load capacity $R^s(n^s)$ of the node n^s in the substrate/physical network is defined as follows:

$$R^s(n^s) = R_N^s(n^s) \cdot \sum_{e^s \in E^s(n^s)} \frac{1}{2} R_E^s(e^s)$$

Based on this, the load balancing capacity $\overline{R^s(n^s)}$ of the node n^s in the substrate/physical network is defined as follows:

$$\overline{R^s(n^s)} = w_1 \cdot \overline{R^s(n^s)} + w_2 \cdot \sum_{m^s \in N^s(n^s)} \frac{R^s(n^s)}{\sum_{h^s \in N^s(m^s)} R^s(h^s)} \cdot \overline{R^s(m^s)}$$

Namely, the load balancing capacity of the node is related to the self-provided load balancing capacity of the node and the load balancing capacity of the neighbor node thereof, wherein w_1, w_2 are relevant weight affecting factors, which satisfy $w_1 + w_2 = 1$,

and $\frac{R^s(n^s)}{\sum_{h^s \in N^s(m^s)} R^s(h^s)}$ is the effect of the node m^s on the node n^s . Therefore,

the load balancing matrix A_i^s of the area i may be defined as follows:

$$A_i^s(m^s, n^s) = \begin{cases} w_1 & \text{if } m^s = n^s, m^s \in N_i^s \\ \frac{w_2 \cdot R^s(n^s)}{\sum_{h^s \in N^s(m^s)} R^s(h^s)} & \text{if } (m^s, n^s) \in E_i^s \\ 0 & \text{other} \end{cases}$$

The load balancing capacity of the node in the area i is expressed as follows:

$$\overline{R_i^s} = \overline{R_i^s} \cdot A_i^s$$

Similarly, the load balancing demand $\overline{R^v}$ of the node in the virtual network request may be defined as: $\overline{R^v} = \overline{R^v} \cdot A^v$, wherein the load demand $R^v(n^v)$ of the virtual node in the virtual network request satisfies:

$$R^v(n^v) = R_N^v(n^v) \cdot \sum_{e^v \in E^v(n^v)} \frac{1}{2} R_E^v(e^v)$$

the load balancing demand $\overline{R^v(n^v)}$ of the virtual node n^v satisfies:

$$\overline{R^v(n^v)} = w_1 \cdot \overline{R^v(n^v)} + w_2 \cdot \sum_{m^v \in N^v(n^v)} \frac{R^v(n^v)}{\sum_{h^v \in N^v(m^v)} R^v(h^v)} \cdot \overline{R^v(m^v)}$$

Namely, the load balancing demand of the virtual node is related to the self-provided load balancing demand of the load balancing demand of the node and the

neighbor node thereof, wherein $\frac{R^v(n^v)}{\sum_{h^v \in N^v(m^v)} R^v(h^v)}$ is the effect of the node m^v on the node n^v .

The load balancing matrix A^v of the virtual network request is defined as follows:

$$A^v(m^v, n^v) = \begin{cases} w_1 & \text{if } m^v = n^v, m^v \in N^v \\ \frac{w_2 \cdot R^v(n^v)}{\sum_{h^v \in N^v(m^v)} R^v(h^v)} & \text{if } (m^v, n^v) \in E^v \\ 0 & \text{other} \end{cases}$$

Therefore, the load balancing demand $\overline{R_i^v}$ of the node in the i virtual network

subrequest may be defined as: $\overline{R_i^v} = \overline{R_i} \cdot A_i^v$.

In order to explain the objects, technical solutions and advantages of the invention more clearly, the technical solutions in the embodiments of the invention will be described clearly and completely with reference to the drawings in the embodiments of the invention hereinafter. Apparently, the embodiments described are merely partial embodiments of the invention, rather than all embodiments.

As shown in Fig. 1, the working flow chart of a global manager in a classification-based virtual network embedding method provided by the invention is as follows: when starting, the global manager waits for the arrival of a virtual network request; 2) when the virtual network request arrives, the global manager performs classification analysis on virtual nodes in the virtual network request and determines classification and area to which each virtual node belongs; 3) divides the virtual network request into a plurality of virtual subrequests according to the classification results; 4) allocates the virtual network subrequests to each local manager to process respectively; and 5) when each local manager finishes processing, combines and feeds back the processing results of each local manager to users.

When performing classification analysis on the virtual nodes in the virtual network request, demand limit like the geographical position of the virtual node is mainly considered, and the virtual nodes are classified to the substrate/physical network in the substrate/physical network of an area nearest to the geographical position required

thereof; therefore, the classification $\phi(n^v) = \arg \min_{i \in [0, |O^v|)} \{dist(L^v(n^v), O_i^v)\}$ to which the virtual node n^v belongs may be defined.

The virtual nodes belonging to different classifications are divided to different virtual network subrequests according to the classification results of the virtual network request, and links between the subrequests are simplified. A virtual node n^v belonging to the i virtual network subrequest $G_i^v(N_i^v, E_i^v)$ satisfies $\phi(n^v) = i$, all virtual links spanning the i virtual network subrequest and the j virtual network subrequest are

simplified into one link $e(i, j)$, which satisfies the conditions:

$$R_E^v(e(i, j)) = \max_{m^v \in N_i^v, n^v \in N_j^v} \{R_E^v(m^v, n^v)\}$$
, i.e., the normalization demand of the simplified virtual link is the maximum of the normalization demand of all the links spanning the i virtual network subrequest and the j virtual network subrequest; if the virtual network embedding result can satisfy the demand of the simplified link $e(i, j)$, then the demands of all the virtual links spanning the i virtual network subrequest and the j virtual network subrequest will be satisfied. After the virtual network request is divided into a plurality of virtual network subrequests, each virtual network subrequest is allocated into the substrate/physical network of a proper area, and the virtual network subrequest i , namely $G_i^v(N_i^v, E_i^v)$, will be allocated to the i substrate/physical network for further mapping.

Fig. 3 is a block diagram of a substrate/physical network framework in an embodiment of the classification-based virtual network embedding method provided by the invention. As shown in Fig. 3, the substrate/physical network in the embodiment is a two-layer structure, i.e., a management layer and a service layer, wherein the management layer includes a global manager and a local manager. The local manager collects node information and link state in the area, and calculates the position center of the area and the logic topology of the node in the area, and then sends the updated calculation results to the global manager. The global manager is communicated with the local manager directly to keep the latest state of the entire physical network, wherein the local manager regularly collects each the information of each node and link in the area. The local manager sends update information to the global manager initiatively only when the network state in the area changes so as to reduce communication expenses. In addition, the global manager passively receives the network state information of the area in general case, but may still enquiry the local manager initiatively and ask for related information.

Fig. 4 is a block diagram of virtual network embedding of the global manager in

the embodiment of the classification-based virtual network embedding method provided by the invention. As shown in Fig. 4, when the virtual network request VNR arrives, the global manager firstly performs classification analysis on the topology of the virtual network request, and divides the virtual network request into small virtual network subrequests VNR_{sub1} and VNR_{sub2} according to the classification results. The virtual network subrequests are allocated to the substrate/physical network in a proper area for parallel and centralized virtual network embedding according to the classification to which each virtual network subrequest belongs. As shown in Fig. 4, the virtual network subrequest VNR_{sub1} is allocated to area 1 for mapping, and the virtual network subrequest VNR_{sub2} is allocated to area 3 for mapping.

As shown in Fig. 2, the working flow chart of the local manager in the classification-based virtual network embedding method provided by the invention is as follows: 1) when starting, the local manager calculates the load balancing capacity of each node in the bottom network node in the area; waits for the arrival of a virtual network subrequest; 3) calculates the load balancing demand of each virtual node in the virtual network subrequest when the virtual network subrequest arrives; and 4) maps the virtual network subrequest on the substrate/physical network in the area.

When calculating the load balancing capacity of the nodes of the substrate/physical network in the area and the load balancing demand of the virtual network subrequest, the foregoing load balancing mechanism will be utilized for calculation. For the node of the substrate/physical network in the area i , the load balancing capacity thereof is represented as $\overline{R}_i^s = \overline{R}_i^s \cdot A_i^s$, and an algorithm pseudocode thereof is as follows:

$\overline{R}_i^s = RBlance(A_i^s, \lambda, \delta)$ Algorithm 1 the load balancing

capacity $\overline{R}_i^s = RBlance(A_i^s, \lambda, \delta)$ of bottom network node in the area

Input:

A_i^s : the load balancing matrix of the area i

	λ : maximum iterative calculation times
	δ : when the error of two iterations is less than δ , stop iteration
	Output:
	$\overline{R}_i^s$: the load balancing capacity of each node of the substrate/physical network in the area
:	$\overline{R}_i^s = RBlance(A_i^s, \lambda, \delta)$
2:	$k \leftarrow 0, \overline{R}_i^0 \leftarrow R_i^s(n^s)$
3:	while $k < \lambda$ do
:	$\overline{R}_i^{k+1} \leftarrow \overline{R}_i^k \cdot A_i^s$
:	if $\ \overline{R}_i^{k+1} - \overline{R}_i^k\ < \delta$ then break
:	$k \leftarrow k + 1$
:	end while

The load balancing demand $\overline{R}_i^v = \overline{R}_i^s \cdot A_i^v$ of the i virtual network subrequest is calculated using the same algorithm in the present invention.

Fig. 5 is a block diagram of a load balancing mechanism in the embodiment of the classification-based virtual network embedding method provided by the invention. The load balancing demand and the load balancing capacity are respectively calculated for the virtual network subrequest and the substrate/physical network in the area according to the present invention. As shown in Fig. 5, the load balancing mechanism is utilized to

calculate the load balancing capacity of the substrate/physical network according to the present invention, and the calculation of the load balancing demand of the virtual network subrequest is similar. The calculation of the load balancing demand and the load balancing capacity depend on the load balancing matrix of each network topology (as shown in Table 4) and the load capacity of each node (as shown in Table 3). It may be seen from the calculation result of table 5 that the node B in the area has the maximum load balancing capacity and the node G has the minimum load balancing capacity.

Table 3 Load capacity of each node

A	B	C	D	E	F	G
175	387	85	247	90	160	72
			.5			

Table 4 Load balancing matrix

	A	B	C	D	E	F	G
A	0.7 0	0.1 7	0	0	0	0.0 9	0.0 4
B	0.0 9	0.7 0	0.0 4	0.1 2	0.0 5	0	0
C	0	0.1 6	0.7 0	0.1 4	0	0	0
D	0	0.1 9	0.0 6	0.7 0	0.0 6	0	0
E	0	0.1 2	0	0.1 1	0.7 0	0.0 7	0
F	0.1 6	0	0	0	0.0 8	0.7 0	0.0 6
G	0.1	0	0	0	0	0.1	0.7

	6					4	0
--	---	--	--	--	--	---	---

Table 5 Load balancing capacity of each node

A	B	C	D	E	F	G
182	344	91.	229	125	108	48.
.5	.3	19	.5	.5	.4	48

In the present invention, a local management module employs a greedy strategy to map the virtual network subrequests onto the bottom physical layer network in the area according to a load balancing demand of each node in the virtual network subrequest and the load balancing capacity of each node in the area. During the mapping process of the present invention, the mapping of the virtual nodes and the mapping of the virtual links are performed at the same time. That is, when mapping the virtual nodes, the demand characteristics of the virtual links are combined so as to avoid expenses of a longer path in the substrate/physical network. During mapping, a virtual node n^v which is unmapped currently and has maximum load balancing demand in the virtual network subrequest is firstly detected, then k candidate bottom physical nodes of the virtual node n^v in the area are sought, wherein the k candidate nodes are the first n^v bottom physical nodes with maximum load balancing capacity which are unmapped in the area currently and can satisfy the demand of the virtual node n^v . For each candidate bottom physical node, the sum of the path length between the candidate bottom physical node and the bottom physical nodes is calculated corresponding to each mapped virtual node, the virtual node n^v is mapped on the candidate node having shortest path length, and the virtual links between the mapped virtual nodes are mapped.

A mapping algorithm pseudocode of the local manager to the virtual network subrequest is described as follows:

Algorithm 2 the mapping algorithm of the virtual network

subrequest in the area $success = CoordinateMap(\overline{R_i^s}, \overline{R_i^v}, k)$

Input:

$\overline{R_i^s}$: the load balancing capacity of each node in the substrate/physical network in the area

$\overline{R_i^v}$: load balancing demand of each node in the i virtual network subrequest

k : number of the candidate bottom physical nodes

Output:

$success$: whether the mapping result is successful or not

:	$success = CoordinateMap(\overline{R_i^s}, \overline{R_i^v}, k)$
:	Q_{mapped}^s stores substrate nodes that host virtual nodes in N_i^v with unmapped links
:	sort virtual and substrate nodes in N_i^v and N_i^s according to $\overline{R_i^v}$ and R_i^s
:	for $m = 0$ to $ N_i^v $ do
:	if $Q_{mapped}^s.size == 0$ then
:	$n_{i_{arg\ est}}^s = findCandidate(N_i^v[m], 1)$
:	else
:	$Q_{candidate}^s \leftarrow findCandidate(N_i^v[m], k)$
:	$P_{candidate}^s \leftarrow computePath(Q_{candidate}^s, Q_{mapped}^s)$

0:	$n_{target}^s \leftarrow findNodeWithShortestPath(P_{candidate}^s)$
1:	end if
2:	if $mapNode(n_{target}^s, N_i^v[m])$ then
3:	map path between mapped nodes
4:	$push(Q_{mapped}^s, n_{target}^s)$
5:	else
6:	return false
7:	end if
8:	end for

Figs. 6-1, 6-2 and 6-3 are mapping block diagrams of a virtual network of a local manager in the embodiment of the classification-based virtual network embedding method provided by the present invention. Numbers in small boxes near each node in the three figures above represent normalized resources of the nodes which are affected by storage capacity, magnetic disk I/O, CPU, memory size and the like; and numbers on lines represent normalized resources of links which are affected by band width, delay and jitter; moments (t1, t2, ...) on wavy lines represent processing orders of directed nodes or links during mapping; and numbers in "  " represent the load balancing capacity/load balancing demand of the node. In the embodiment, the number of the candidate nodes $k=3$. As shown in Fig. 6-2, before mapping, the node having

maximum load balancing demand in the virtual network subrequest is a, and the node having maximum load balancing capacity in the substrate/physical network in the area is B at the same time, wherein the resource demand of the node a is 12, and the available resources of the node B is 14; therefore, the node B can satisfy the demand of the node a, and the node a is mapped into the node B. At this moment, the node with the maximum load balancing demand which is unmapped in the virtual network subrequest is the virtual node c, and the candidate nodes of the virtual node c in the substrate/physical network are D, A and E (the reason is: the number of the candidate nodes $k=3$, excluding the mapped bottom physical node B, the rest three candidate nodes with maximum load balancing capacity are D, A and E), wherein the path lengths from the three nodes D, A and E to the mapped node B are all 1, while D has the maximum load balancing capacity; therefore, the virtual node c is mapped on D, then a virtual link $\langle a, c \rangle$ between the mapped virtual nodes a and c is mapped on a physical link $\langle B, D \rangle$. At this moment, the virtual node to be mapped is b, and the candidate nodes of the virtual node b in are A, E and F, wherein the path length from the candidate node A to the mapped node is 4, the path length of the candidate node E to the mapped node is 2, and the path length of the candidate node F to the mapped node is 4; therefore, the virtual node b is mapped on the candidate node E so as to reduce the mapping expenses of the longer path, then a virtual link $\langle a, b \rangle$ is mapped on a bottom physical path $\langle B, E \rangle$, and a virtual link $\langle c, b \rangle$ is mapped to a bottom physical path $\langle D, E \rangle$.

In the classification-based virtual network embedding method of the present invention, when all the virtual network subrequests are successfully mapped to the substrate/physical networks in each area, the entire virtual network request is successfully mapped. Otherwise, mapping and resource allocation are rejected, and results are returned to the users.

In conclusion, the computation complexity is decreased and the mapping time is reduced by the present invention through performing classification analysis on the virtual nodes in the virtual network request, dividing the virtual network of the request into small virtual network subrequests according to the classification results, and allocating the virtual network subrequests to each local manager to perform parallel

mapping. When each local manager is mapping the virtual network subrequest, the load balancing demand of the virtual network subrequest and the load balancing capacity of the substrate/physical network are calculated for helping mapping by means of introducing the resource balancing mechanism, thus implementing the load balancing of the shared bottom resource. In the local manager, the coordination and integration of the mapping of the virtual nodes and the mapping of the virtual links avoid wide distribution of mapping of the nodes, and reduce the bottom physical path length of mapping of the links.

It should be finally noted that the above embodiments are for purposes of illustration and description to the technical solution of the present invention merely, but are not intended to limit the present invention. Although detailed illustration is made to the present invention with reference to embodiments, it may be understood by those having ordinary skills in the art that modification or equivalent replacement may be performed on the technical solution of the present invention without departing from the spirit and scope of the technical solution of the present invention, which shall all fall within the scope of the claims of the present invention.

Claims

1. A classification-based virtual network embedding method, wherein the method comprises:

step 101) when receiving a virtual network request, classification virtual nodes related to a virtual network request according to the classification standard, and dividing the virtual nodes belonging to different classifications to different virtual network subrequests;

wherein the classification standard is namely to classify the virtual nodes into a substrate/physical network of an area nearest thereof according to the demand information of the virtual nodes on geographical positions; and all the virtual nodes in the same virtual network subrequest belong to the same classification, and the classification is namely a classification to which any virtual node in the virtual network subrequest belongs;

step 102) sending each virtual network subrequest to a certain area in the substrate/physical network to which the virtual network subrequest belongs to; and

step 103) mapping the virtual network subrequests onto the bottom physical layer network in the area according to a load balancing demand of each node in the virtual network subrequest and the load balancing capacity of each node in the area; wherein,

the load balancing demand represents the load demand of each virtual node in case of balanced loads; and the load demand is defined as a product of resources of the virtual node request and resources of all link requests connected to the virtual node; and

the load balancing capacity represents the load capacity of each physical node in the area in case of balanced load; and the load capacity is defined as a product of available resources of the physical node and available resources of all links connected to the physical node;

the load balancing demand is obtained by employing the following method:

step 301 respectively performing normalization processing on the virtual node and virtual link demands in each virtual network subrequest;

wherein, the normalization processing of the virtual nodes is related to the

demands on the storage capacity of the nodes, quantity of magnetic disk I/O and CPU, and memory size; and the normalization processing of the virtual links is related to the demands on the band width, delay and delay jitter of the links;

step 302) calculating the load demand of each virtual node in each virtual network subrequest;

wherein, the load demand is defined as a product of demands of the virtual node and demands of all links connected to the virtual node;

step 303) calculating the load balancing matrix of each virtual network subrequest;

wherein, the elements of the load balancing matrixes are: a weighted value mutually affected by the load balancing demands between the virtual nodes in the virtual network subrequest, and the weighted value is particularly as follows: when the virtual node n^v is just the virtual node m^v , i.e., $n^v = m^v$, or the virtual node n^v and the virtual node m^v are connected by a virtual link, the weighted value is defined to be a nonzero positive number; otherwise, the weight affected by n^v to the load balancing demand of m^v is 0;

wherein, n^v represents the n^v line of the matrix, m^v represents the m^v row of the matrix, both the line number and the row number of the matrix are the total number of the virtual nodes, and a corner mark v represents the virtual network request; and

step 304) iteratively calculating the load balancing demand of each virtual node in the area according to the load balancing matrix, i.e., the load balancing demand of the virtual node n^v being the sum of the load balancing demands affected by all the virtual nodes in the virtual network subrequest.

2. The classification-based virtual network embedding method according to claim 1 wherein step 103) further comprises:

step 103-1) detecting a virtual node n^v which is unmapped currently and has

maximum load balancing demand in the virtual network subrequest;

step 103-2) seeking k candidate bottom physical nodes of the virtual node n^v in the area, wherein the k candidate nodes are the first k bottom physical nodes with maximum load balancing capacity which are unmapped in the area currently and can satisfy the demand of the virtual node n^v ; and

step 103-3) for each candidate bottom physical node, calculating the sum of the path length between the candidate bottom physical node and the bottom physical nodes corresponding to each mapped virtual node, mapping the virtual node n^v on the candidate node having shortest path length, and mapping the virtual links between the mapped virtual nodes.

3. The classification-based virtual network embedding method according to claim 1 or 2, wherein the load balancing capacity is obtained as follows:

step 201) respectively performing normalization processing on resources of the physical nodes and physical links of each area inclusive in the substrate/physical network;

wherein, the normalization processing of the physical nodes is related to the storage capacity of the nodes, magnetic disk I/O and quantity of CPU, and memory size; and the normalization processing of the physical links is related to the bandwidth, delay and delay jitter of the links;

step 202) calculating the load capacity of the nodes in each area in the substrate/physical network;

wherein, the load capacity is defined as a product of available resources of the physical node and available resources of all links connected to the physical node.

step 203) calculating load balancing matrix of each area in the substrate/physical network;

wherein, the elements of the balanced load matrix are: a weighted value mutually contributed by the load balancing capacities between the physical nodes in a certain area, and the weighted value mutually contributed by the load balancing capacities is

particularly as follows: when the physical node n^s is just the physical node, i.e., $n^s = m^s$, or the physical node n^s and the physical node m^s are connected by a physical link, the physical node n^s contributes to the load balancing capacity of the physical node m^s , and the weighted value is a nonzero positive number at the moment; otherwise, the weight contributed by the physical node n^s to the load balancing capacity of the physical node m^s is 0; wherein, n^s represents the n^s line of the matrix, m^s represents the m^s row of the matrix, both the line number and the row number of the matrix are the total number of the physical nodes, and a corner mark s represents the substrate/physical network; and

step 204) iteratively calculating the load balancing capacity of each physical node in the area according to the load balancing matrix, i.e., the load balancing capacity of the physical node n^s is the sum of the load balancing capacities contributed by all the physical nodes in the area.

4. A classification-based virtual network embedding system, wherein the system comprises:

a global management module configured to, when receiving a virtual network request, classify virtual nodes related to the virtual network request according to a classification standard, and divide the virtual nodes belonging to different classifications to different virtual network subrequests, wherein the classification standard means that one virtual node classifies the virtual nodes into a substrate/physical network of an area nearest thereof; and send each virtual network subrequest to a certain area in the substrate/physical network to which the virtual network subrequest belongs to; wherein when classification analysis is performed on virtual nodes in a virtual network request, the virtual nodes are classified into a substrate/physical network of an area nearest thereof according to the demand information of the virtual nodes on geographical position; and

a local management module configured to map the virtual network subrequests

onto the bottom physical layer network in the area according to a load balancing demand of each node in the virtual network subrequest and the load balancing capacity of each node in the area;

wherein, the load balancing demand represents the load demand of each virtual node in case of balanced loads; and the load demand is defined as a product of resources of the virtual node request and resources of all link requests connected to the virtual node;

the load balancing capacity represents the load capacity of each physical node in the area in case of balanced load; and the load capacity is defined as a product of available resources of the physical node and available resources of all links connected to the physical node; the local management module further comprises:

a load balancing capacity or load balancing demand acquisition submodule configured to acquire the load balancing capacity value of each node in each area of the substrate/physical network, and acquire the load balancing demand value of each virtual node of each virtual network subrequest;

a detection submodule configured to detect a virtual node n^v which is unmapped currently and has maximum load balancing demand in the virtual network subrequest;

a candidate bottom physical node selection submodule configured to seek k candidate bottom physical nodes of the virtual node n^v in the area, wherein the k candidate nodes are the first k bottom physical nodes with maximum load balancing capacity which are unmapped in the area currently and can satisfy the demand of the virtual node n^v ; and

a node and link mapping submodule configured to, for each candidate bottom physical node, calculate the sum of the path length between the candidate bottom physical node and the bottom physical nodes corresponding to each mapped virtual node, map the virtual node n^v on the candidate node having shortest path length, and map the virtual links between the mapped virtual nodes.

5. The classification-based virtual network embedding system according to claim 4, wherein the load balancing capacity acquisition submodule further comprises:

a first normalization processing module configured to respectively perform normalization processing on resources of the physical nodes and physical links of each area inclusive in the substrate/physical network;

wherein, the normalization processing of the physical nodes is related to the storage capacity of the nodes, magnetic disk I/O and quantity of CPU, and memory size; and the normalization processing of the physical links is related to the band width, delay and delay jitter of the links;

a first node load capacity acquisition module configured to calculate the load capacity of each node in the substrate/physical network;

wherein, the load capacity is defined as a product of available resources of the physical node and available resources of all links connected to the physical node;

a load balancing matrix acquisition module configured to calculate the load balancing matrix of each node in the substrate/physical network;

wherein, the load balancing matrix defines a weighted value mutually contributed by the load balancing capacities between the physical nodes in the area of the substrate/physical network, when and only when the physical node n^s is just the physical node, i.e., $n^s = m^s$, or the physical node n^s and the physical node m^s are connected by a physical link, the weighted value is a nonzero positive number at the moment; otherwise, the weight contributed by the physical node n^s to the load balancing capacity of the physical node m^s is 0;

a load balancing capacity calculation module configured to iteratively calculate the load balancing capacity of each physical node in the area according to the load balancing matrix, i.e., the load balancing capacity of the physical node n^s is the sum of the load balancing capacities contributed by all the physical nodes in the area.

6. The classification-based virtual network embedding system according to claim 4, wherein the load balancing demand acquisition submodule further comprises:

a second normalization processing module configured to respectively perform normalization processing on the virtual node and virtual link demands in each virtual network subrequest;

wherein, the normalization processing of the virtual nodes is related to the demands on the storage capacity of the nodes, quantity of magnetic disk I/O and CPU, and memory size; and the normalization processing of the virtual links is related to the demands on the band width, delay and delay jitter of the links;

a second load demand calculation module configured to calculate the load demand of each virtual node in each virtual network subrequest;

wherein, the load demand is defined as a product of demands of the virtual node and demands of all links connected to the virtual node;

a second load balancing matrix calculation module configured to calculate the load balancing matrix of each virtual network subrequest;

wherein, the load balancing matrix defines a weighted value mutually affected by the load balancing demands between the virtual nodes in the virtual network subrequest, when and only when the virtual node n^v is just the virtual node m^v , i.e., $n^v = m^v$, or the virtual node n^v and the virtual node m^v are connected by a virtual link, the weighted value is a nonzero positive number at the moment; otherwise, the weight affected by the virtual node n^v to the load balancing demand of the virtual node m^v is 0; and

a load balancing demand acquisition module configured to iteratively calculate the load balancing demand of each virtual node in the area according to the load balancing matrix, i.e., the load balancing demand of the virtual node n^v being the sum of the load balancing demands affected by all the virtual nodes in the virtual network subrequest.