Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional Failures

Hongfang Yu[*], Chunming Qiao[†], Vishal Anand[#], Xin Liu[^], Hao Di[*], Gang Sun[*]

[*]School of Communication and Information Engineering, University of Electronic Science and Technology of China, China
[†]Department of Computer Science and Engineering, State University of New York at Buffalo, USA
[#]Department of Computer Science, The College at Brockport, State University of New York, USA
^Brookhaven National Laboratory, USA

Abstract-**As virtualization becomes more and more popular, how to guarantee survivability of a virtual infrastructure (VI) over a wide-area optical network is increasingly important. In this paper, we approach the problem of survivable VI mapping (SVIM) from a few unique perspectives. One of the most distinguishing perspectives is that a large-scale regional failure could destroy one or more facility nodes to which some VI nodes are mapped. Accordingly, redundant facility nodes at different geographical locations and redundant optical connections have to be provisioned such that the VI can still be mapped after the failure. Another distinguishing perspective is that with failure-dependent protection, the SVIM problem can be decomposed into several instances of the basic non-survivable VI mapping (NSVIM) problem, whose solution permits effective sharing of the redundant resources among all failures.**

In this paper, we first formulate the minimum-cost SVIM problem using mixed integer linear programming (MILP). We then propose an efficient heuristic solution to NSVIM, based on which two novel heuristic SVIM algorithms called Separate Optimization with Unconstrained Mapping (SOUM) and Incremental Optimization with Constrained Mapping (IOCM). Simulations are performed to study and compare the performance of the MILP and heuristics.

I. Introduction

Emerging distributed applications and services require the coordination of multiple geographically distributed computing resources that are networked together. Examples of such applications include nationwide (or even worldwide) cloud computing [1], large-scale simulation and peta-scale scientific experiments [2].

Two key technologies enable such large scale distributed applications. One is *virtualization* in the computing regime, which refers to a broad range of approaches for the abstraction of computing resources at different levels. The other technology is high-bandwidth *optical networks* [3] as the substrate.

With the maturity of the virtualization and optical networking technology, in the near future we will be able to deploy a Federated Computing and Networking System (FCNS), which interconnects a large number of computing facility nodes (e.g., clusters and data centers) with optical networks, to support a variety of distributed applications and services.

In this work, we refer to either an application or a service request submitted to a FCNS as a Virtual Infrastructure (VI) request, which can represent a large portion of emerging distributed applications and services. A VI request consists of a set of VI nodes, with each node requiring some computing resources (e.g., CPU resource) on a separate computing facility node (hereby called facility node for short). A VI node also needs to communicate with another VI node to send intermediate results, file data, or some other information. As a result, a VI request imposes connectivity requirements among the VI nodes in terms of topology, bandwidth, and delay guarantees. We assume that each VI node may be mapped to any facility node with enough available required computing resources. Thus, given a VI request, a FCNS has to find a mapping of the VI request onto the FCNS substrate. That is, assign each VI node to a facility node and establish communication between these facility nodes. A similar problem has been studied in the context of network virtualization to overcome Internet ossification [5-7]. However, the work in [5,6] do not consider any computing or bandwidth constraints on the substrate network while mapping the VI requests. Moreover, none of these works consider VI request survivability in the event of substrate network failures.

For mission-critical VI requests, it is essential to maintain functionality even in the event of failure(s). In this work, we consider regional failure(s) [4]. In general, a region refers to a geographic region of nodes and links that may fail simultaneously due to events such as natural disasters (e.g., earthquakes) or intentional attacks (e.g., bombs). Such a failure will affect the facility nodes, which in turn will affect the VI nodes mapped onto these facility nodes as well as the links which connect these VI nodes. In order to survive from the disruptions due to any single regional failure, the FCNS must reserve redundant facility nodes and bandwidth on fiber links such that after a regional failure, there are enough remaining computing and networking resources that can be used to remap the VI request. Thus, this work encompasses works on *both* fault-tolerant computing [9] and optical network survivability [3,8]. The authors of [10] addressed survivable VI request provisioning and proposed two *failure-independent* approaches cluster and path protection (CPP) and virtual network protection (VNP). However, these failure-independent approaches generally require more redundant resources. The authors of [11] propose a hybrid policy solution to incorporate single substrate link failure the policy is based on a fast rerouting strategy and utilizes a pre-reserved quota for backup on each physical link. This hybrid policy is essentially a restore approach, and can't guarantee 100% recovery. In this paper, we adopt *failure-dependent* protection approach whereby there is a backup solution associated with each regional failure scenario with the objective of minimizing the redundant resources/cost.

To the best of our knowledge, this is the first work that addresses the problem of failure-dependent survivable mapping of VI request in a FCNS. A key challenge of the survivable VI mapping (SVIM) problem is the effective and joint allocation of computing and networking resources. We formulate the SVIM

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

problem as an optimization problem and propose two heuristic algorithms called separate optimization with unconstrained mapping (SOUM) and incremental optimization with constrained mapping (IOCM) to solve the survivable VI request mapping problem.

The remainder of this paper is organized as follows. Section II describes our network model and the survivable mapping problem. Section III formulates the survivable request mapping problem as an MILP. Section IV describes the heuristic algorithms and V presents the performance evaluation of the proposed approaches. Section VI concludes the paper.

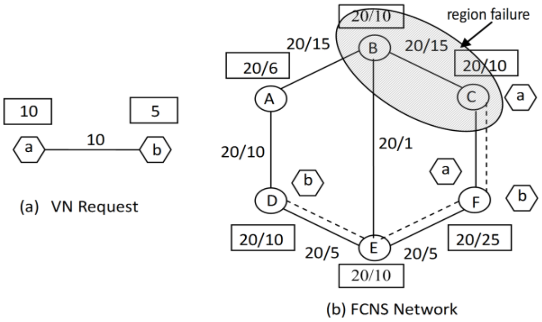
II. Network Model and Problem statement

A. FCNS Substrate

We model the FCNS as an undirected graph $G_S=(V_S,E_S) = (V_F \cup V_X,E_S)$, where V_S is the set of substrate nodes, and E_S corresponds to the set of bidirectional fiber links. Note that V_S composes of V_F and V_X. V_F is the set of facility nodes; V_X is the set of optical switches. We assume that each node in V_F may be connected to one or more switches in V_X. In addition, each failure may either destroy a substrate node v_F in V_F, or completely disconnect it from the rest of network by destroying the links connecting substrate nodes in V_X to v_F. In either case, any VI node that was mapped to such a substrate node v_F will have to be remapped to another location. We focus only on the failure of facility nodes in V_F. Accordingly, to simplify the discussion, we will ignore V_X (i.e., $V_S=V_F$). Assume that the number of VI nodes is N, the number of VI links is M.

Each facility node $v_F \in V_F$ that can provide computing resources, is associated with a weight $c(v_F)$ that represents the available capacity of computing resources at facility node v_F. The cost of a computing resource unit on facility node v_F is $cf(v_F)$. For each link $e_S \in E_S$, the available bandwidth capacity is $w(e_s)$ and the cost of a unit of bandwidth is $cl(e_s)$.

Fig.1 (b) shows a substrate network, where the numbers over the links represent the available bandwidth and the cost of a bandwidth unit, and the numbers in the rectangles represent available computing resources and cost of a computing resource unit.



(a) VN Request

(b) FCNS Network

Fig.1 Mapping of VI Request onto a substrate network

B. VI Request

A VI request (with QoS requirements) is also modeled as an undirected graph $G_L= (V_L, E_L)$, where V_L corresponds to the set of VI nodes, and E_L denotes the set of bidirectional communication demands among the VI nodes.

Each VI node $v_L \in V_L$ needs a certain amount of computing resources for its execution, denoted by $\varepsilon(v_L)$, which represents the amount of resources required by v_L. Each communication demand $e_L \in E_L$ has a bandwidth requirement, denoted by $b(e_L)$. Fig.1 (a) shows one VI request with two VI nodes and a VI link, and associated computing and communication resource requirements.

C. Regional Failures

In [12] the authors showed that there exist only a finite number of distinct regional failures in a given geographical area. Accordingly, we assume a set of possible regional failures R is given. We assume that each regional failure $r \in R$ will simultaneously destroy one or more facility nodes, denoted by $G_r=(V_r, E_r)$, where $V_r \subset V_F$, and $E_r \subset E_S$. In fact we can generalize a regional failure to the concept of a shared risk link group (SRLG) failures. The approach to be discussed in the paper is applicable to the generalized case.

D. Survivable VI Mapping under Single Regional Failures

Mapping a VI request G_L over a FCNS G_S with survivability against any single regional failure r consists of five parts: initial VI request mapping, redundant facility nodes allocation, redundant link allocation, VI request check-pointing and VI request migration.

Initially, without the consideration of possible failures, we need to allocate a separate facility node and required computing resources for each VI node of the VI request, as well as set up paths and reserve required bandwidth to transmit data among the VI nodes. Assume that the number of VI nodes is N_I, and then exactly N_I facility nodes are allocated in this initial request mapping.

In our failure-dependent SVIM approach, when we consider a regional failure, if a facility node initially allocated for a VI node is within the failed region, we need at least another facility node outside the failed region in order to restore the affected VI node. To ensure that such a facility node can be found in the event of a regional failure, we allocate it before a failure actually happens. Since a regional failure may cause multiple facility nodes allocated for initial VI request mapping to fail, a sufficient set of redundant facility nodes have to be pre-allocated.

Similarly, we need to pre-allocate enough redundant bandwidth on the substrate links to support the communication between the VI nodes under any single regional failure. In particularly, if a VI node is to be restored at a pre-allocated spare facility node, we must set up new spare paths between that spare facility node and other surviving facility nodes.

To eliminate the need for starting over the job/computation of a VI node when it is disrupted by a failure, we need to check-point the VI node. More specifically, we periodically send the status of the VI node (e.g., virtual machine image) to spare facility node such that the VI request can be restored from a previous saved state when a regional failure occurs.

Finally, we need to perform VI request migration, i.e., migrate the failed facility nodes to the spare facility nodes.

In Fig. 1 nodes a and b of the VI request are initially (before failure) mapped to facility nodes C and F using the bandwidth on substrate link C-F. The regional failure destroys facility nodes B, C and the substrate link B-C. The FCNS now remaps the VI request such that the VI node a is now mapped onto facility node F and b is mapped to facility node D using the

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

bandwidth on substrate links E-F and D-E, and the jobs on these failed facility nodes are migrated to the newly facility nodes.

E. Problem Statement

Given: a FCNS $G_S = (V_F, E_S)$, a VI request $G_L = (V_L, E_L)$, a list of possible regional failures R.

Question: how to jointly allocate computing and networking resources, including the spare resources, such that the total cost of survivable mapping of the VI request, i.e., sum of computation and communication cost is minimized?

A basic (non-survivable) mapping solution includes: 1) a one-to-one (but not onto) mapping from V_L to V_F; 2) mapping of each e_L in G_L to a path in G_S. For each regional failure r, a survivable mapping solution includes: 1) a one-to-one mapping from V_L to the surviving facility nodes in V_F-V_r; 2) mapping of each e_L in G_L to a path in G_S-G_r.

We use the computing resource matrix C and the bandwidth resource matrix B (Fig. 2(a) and (b)) to calculate the required computing and bandwidth resources. In matrix C, row 0 corresponds to the basic (non-survivable) mapping with no consideration of any regional failure and each subsequent row represents a survivable mapping for each failure region r_i (i=1 to $|R|$). Each column represents the required computing resources for each failure scenario on a substrate facility node. Each element c_{ij} represents the computing resources that must be allocated on facility node j for the i^{th} regional failure. Similarly in matrix B, each column represents the required communication resources for each failure scenario on a substrate facility link k, and each element b_{ik} represents the bandwidth resources required on substrate link k to recover from the i^{th} regional failure.

Since we assume that only one regional failure occurs at any one time, the resources reserved on the facility nodes and fiber links can be shared among the different failure scenarios. Accordingly to ensure survivability under any regional failure, the total computing resources that are allocated on node j, denoted by rn_j, is the maximum of all resources under any failure (i.e., $rn_j = \max_i\{c_{ij}\}$, i=0 to $|R|$), which is the largest value in column j. Similarly, the total bandwidth allocated on link k, denoted by rl_k, is the maximum of all resources under any failure (i.e., $rl_k = \max_i\{b_{ik}\}$, i=0 to $|R|$).

$$\begin{array}{cc} & \begin{array}{cccc}1 & 2 & & |V_F|\end{array} \\ \begin{array}{c}0\\1\\ \\|R|\end{array} & \left[\begin{array}{cccc} c_{01}, & c_{02}, & \cdots, & c_{0N} \\ c_{11}, & c_{12}, & \cdots, & c_{1N} \\ \cdots & & & \\ c_{|R|1}, & c_{|R|2}, & \cdots, & c_{|R|N}\end{array}\right]\end{array} \qquad \begin{array}{cc} & \begin{array}{cccc}1 & 2 & & |E_S|\end{array} \\ \begin{array}{c}0\\1\\ \\|R|\end{array} & \left[\begin{array}{cccc} b_{01}, & b_{02}, & \cdots, & b_{0M} \\ b_{11}, & b_{12}, & \cdots, & b_{1M} \\ \cdots & & & \\ b_{|R|1}, & b_{|R|2}, & \cdots, & b_{|R|M}\end{array}\right]\end{array}$$

(a) (b)

Fig.2 (a) Computing Resource Matrix C. (b) Bandwidth Resource Matrix B.

III. MILP FORMULATION

A. Augmented Substrate Graph Construction

To formulate the cost-minimization SVIM problem, we apply the following graph transformation similar to [7] to G_S. We add N_I "virtual nodes" into G_S; each virtual node corresponds to a VI node in the VI request v_L and is set to have an unlimited computing capacity. Such a virtual node is connected to all the facility nodes $v_F \in V_F$ that has enough available computing resources required by the corresponding VI node v_L. We call the link connecting such a virtual node and facility node v_F as a "virtual link". We assume that the bandwidth resources on each virtual link are unlimited, and that each virtual node is unaffected by any regional failure in R. However, for each virtual link connecting a virtual node with facility node v_F, if v_F is within a region $r \in R$, then that virtual link is also inside the region r.

B. Region-failure-dependant MILP formulation

Based on the above transformation the SVIM problem can now be formulated as a mixed integer multi-commodity flow problem. We consider each communication demand in G_L as a commodity with source and destination nodes s_i and $t_{i,}$ ($\forall i$, s_i, $t_\text{i} \in V_L$) respectively. Each flow starts from a virtual node s_i and ends in another virtual node t_i. In each regional failure r, every virtual node must choose one and only one available virtual link to connect itself to a FCSN substrate node. This constraint ensures that each VI node corresponding to that virtual node is mapped to an available substrate node under each regional failure. At the same time, all the virtual links (i.e., flows) are also mapped efficiently inside the substrate network.

Due to space limitation we omit the detailed MILP formulation and only present the objective function.

<u>Objective function:</u>

$$\sum_{k \in E_s} rl_k * cl_k + \sum_{j \in V_F} rn_j * cf(j) \qquad (1)$$

where rl_k denotes the total required bandwidth resource of fiber link k to support either initial VI mapping or failure tolerance and cl_k is the cost of a bandwidth unit on fiber link k. And rn_j denotes the total required computing resource on the facility node j to support either initial VI mapping or failure tolerance, and cf (j) is the cost of a computing unit on facility node j. Thus objective function (1) tries to minimize the total cost of all facility nodes and fiber links to support either initial VI mapping or failure scenarios.

IV. SURVIVABLE VI PROVISIONING ALGORITHM

Since finding an optimal survivable VI mapping using the MILP is computationally intractable, we propose efficient heuristics to solve the problem.

Our main idea is to start with a "working" mapping of the VI to the FCNS, and then for each failure scenario, derive a "backup" mapping of the VI to only the FCNS nodes and links that will survive the failure. It is worth noting that even the problem of obtaining a minimum-cost working mapping (i.e., the basic NSVIM problem) is NP-hard. Accordingly, we will propose an efficient heuristic algorithm, to be described it next. Such an algorithm can then be used to obtain backup mappings as well. We will discuss how redundant resources needed by these backup mappings can be shared by using two additional heuristic later in the section.

A. Non-Survivable VI Mapping Algorithm (NSVIM)

The NSVIM algorithm satisfies VI requests without any failure survivability requirement. It maps each VI node to a facility node and allocates the requested amount of computing resources on the facility node. It also maps each VI link to one or more fiber links in the FCNS, and allocates the requested bandwidth on the fiber links. The objective is to minimize the total cost including computing cost and communication cost. A pseudo code for the algorithm is shown in Fig.3.

The algorithm uses two sets, MLN and UMLN, to keep track of mapped and unmapped logical (VI) nodes, respectively. In addition, it uses AFN and UAFN to keep track of allocated and unallocated facility nodes, respectively. Clearly, there should be a one-to-one and onto mapping between MLN and AFN as each VI node is mapped to a unique facility node. Initially, both MLN and AFN are empty, UMLN = V_L and UAFN = V_F (note that, as to be described later, the initial settings of these sets can be modified in other variations of NSVIM).

To select a candidate facility node $v_F \in$ UAFN to which the VI node $v_L \in UMLN$ can be mapped, we note that the amount of available computing resources at v_F must be no less than the requested amount [(v_L). In addition, let $Adj(v_L)$ be the subset of VI nodes that are adjacent to v_L in G_L, and the maximum communication bandwidth requested between v_L and each VI node in $Adj(v_L)$ be B(v_L). For the mapping to be feasible, the maximum amount of available link bandwidth on all links adjacent to v_F should be no less than B(v_L). In the proposed NVSIM algorithm, we will first select a candidate set, *S*, of facilitate nodes that meet the above requirements on the available computing and networking (bandwidth) resources, and then choose a mapping that may result in a minimum cost.

```
NSVIM:
1. Initialization: initializes the set of MLN, AFN, UMLN, and
   UAFN. e.g., for finding a working mapping,
   $MLN=AFN=\Phi$, $UMLN=V_L$, $UAFN=V_F$. etc.
2. Sort the VI nodes in UMLN according to their degree.
3. Choose node $v_L$ with the highest degree.
4. Find a subset of candidate facility nodes S in UAFN (as
   discussed above). If $S=\Phi$, return INFEASIBLE.
5. Assign $v_L$ to $v_F$ in S with the minimum mapping cost $C(v_L \rightarrow v_F)$
   (see Eq.2 and related discussion). Reserve [($v_L$)
   computing resource on the facility node $v_F$.
6. For every VI link connecting $v_L$ with $v_L^* \in M\text{-}adj(v_L)$ find
   the minimum-cost path p in FCNS (see discussion
   below), whose every link has available bandwidth
   more than $b(v_L,v_L^{*'})$ . If no such a path can be found,
   return INFEASIBLE. Otherwise, Reserve bandwidth
   $b(v_L,v_L^{*'})$ on every substrate link along the path p.
7. Move $v_L$ from UMLN to MLN, and $v_F$ from UAFN to ANF.
8. If UMLN =$\Phi$, Return SUCCESS. Otherwise, GOTO 2.
```

Fig.3 Pseudo code for the NSVIM algorithm

The key idea of step 5 in the above NSVIM algorithm is to select a facility node in *S* such that the sum of the computing cost and the communication cost associated with the mapping of v_L to v_F, given in Eq.2 below, is minimized.

$$C(v_L \rightarrow v_F) = CPv_F + ACv_F + UACv_F \qquad (2)$$

The first term CPv_F in Eq. 2 is the computing cost, which is equal to $cf(v_F) \times [(v_L)$, and the other two $AC\ v_F$ and $UACv_F$ are the communication cost. More specifically, ACv_F is the communication cost between v_F and a subset of facility nodes in *AFN,* which have been allocated to the VI nodes in set $M\text{-}adj(v_L) = Adj(v_L) \cap MLN$, and $UACv_F$ is the potential (future) communication cost between v_F and a subset of facility nodes in *UAFN,* which might be allocated to the VI nodes in $UM\text{-}adj(v_L) = Adj(v_L) \cap UMLN$.

ACv_F can be calculated as follows. Let $M\text{-}adj(v_F)$ be the set of facility nodes corresponding to $M\text{-}adj(v_L)$. For each node pair (v_F, $v_F^* \in M\text{-}adj(v_F)$), we find the minimum-cost path, and use the sum of the link costs along the path. A total of $|M\text{-}adj(v_F)|$ paths need to be found and the sum of their costs becomes the value of ACv_F.

The rationales behind including $UACv_F$ are 1) since NSVIM is a greedy algorithm that maps one VI node at a time, there is always a risk that a v_F node that looks good now because e.g., it has a lower sum of CPv_F and ACv_F, may turn out to be a bad choice overall, and 2) to avoid choosing such a v_F node, by looking ahead at potential future communication cost to be incurred when additional facility nodes in *UAFN* are allocated to map the VI nodes in $UM\text{-}adj(v_L)$. $UACv_F$ can be *estimated* as follows. For every node pair (v_L, $v_L^* \in UM\text{-}adj(v_L)$),we find a minimum-cost path from v_F to as many $v_F^* \in UAFN$ as possible that can satisfy the bandwidth requirement of $b(v_L,v_L^{*'})$. We then calculate the *average cost* over all such paths as the estimated potential cost to map the virtual link between v_L and $v_L^* \in UM\text{-}adj(v_L)$. If no such a path can be found at all, the average cost is considered infinite (and the v_F is thus not a feasible choice to map v_L). Otherwise, $UACv_F$ is the sum of the average costs, one for each virtual link between v_L and $v_L^* \in UM\text{-}adj(v_L)$.

In the next section we describe two heuristic survivable VI mapping algorithms that are built around NSVIM. We assume that a working mapping is not given a priori. However, our approach also applies if otherwise.

B. Separate Optimization with Unconstrained Mapping (SOUM)

In SOUM, we decompose the SVIM problem into $|R|+1$ separate NSVIM problems, one involving the initial pre-failure (working) mapping of G_L to G_S, and the others involving the after-failure (or backup) mapping of G_L to G_S-G_{ri}, for $1 <= i <= |R|$. More specifically, when using the NSVIM algorithm to determine the after-failure mapping for r_1 for example, one can set UAFN initially to $V_F - V_{r1}$.

For each of these $|R|+1$ NSVIP problems, SOUM tries to minimize the cost associated with the allocation of computing and network resources (by using e.g., the NSVIM algorithm described earlier). In this way, SOUM deals with each regional failure scenario *irrespective* of other scenarios, i.e., for each regional failure, the VI request is remapped without consideration of the initial pre-failure mapping or any other after-failure mappings. Since the failure scenarios are considered separately, the order in which they are considered does not affect the total amount of resources needed for each row in the matrices *B* and *C* mentioned earlier, nor the total cost of the solution to the SVIP problem.

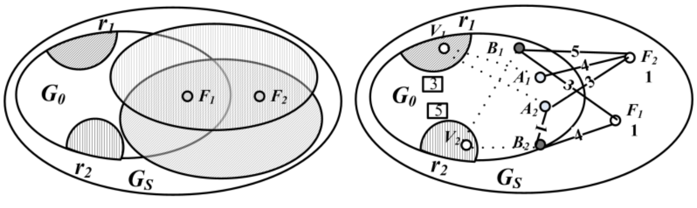
Consider an example of SOUM algorithm in Fig. 4(a), suppose there are two regional failures r_1 and r_2. G_0 is the pre-failure mapping solution, and G_1 and G_2 are the mappings for regional failures r_1 and r_2, respectively. Assuming that 8 units of computing resources are already allocated at facility node F_1 in G_0. However, these 8 units will be ignored when dealing with r_1. That is, when mapping a VI node requiring 10 units of computing resources to F_1 in G_1 using the NSVIM algorithm, 10 units of computing resources will be counted towards the total cost as a part of Eq (2). Similarly, even if 20 units are allocated at F_2 in G_1, when mapping a VI node requiring 16 units to F_2 in G_2, that 16 units will be counted towards the total cost. Of course, after all $|R|$+1 mappings are found, the computing and bandwidth resources required on the substrate network will be the maximum amount of resources required by any failure scenario, as mentioned earlier.

C. Incremental Optimization with Constrained Mapping (IOCM)

This strategy also starts with the initial pre-failure mapping and considers the $|R|$ regional failures one by one. One major difference from SOUM is that each time, it tries to minimize the *additional* computing and networking resources required to deal with the current failure under consideration. In addition, IOCM does not change the mapping for the VI nodes that are unaffected by the current failure. In other words, it only finds additional facility nodes to remap the affected VI nodes.

Accordingly, if $r_1,\ldots,r_{|R|}$ are the order of regional failures, then how G_i is selected will depend on how $G_0\ldots G_{i-1}$ have been selected in the past. That is, the order of failures considered affects the mapping outcome and cost. If $|R|$ is not too large, we can compute the cost and mapping for all possible orderings ($|R|!$) and choose the lowest cost mapping as our final solution. Note that, it is often beneficial not to start first with G_0. For example, if r_1 is the only failure possible, we only need the resources allocated for G_1 (before or after the failure). Accordingly, in our simulations, we will examine all possible orderings of ($|R|$+1) mappings (and there are ($|R|$+1)! of them).

Fig. 4(b) illustrates the basic idea of IOCM. Assume that the order of regional failures considered is r_1 and then r_2. For simplicity, we assume that with respect to G_0, regional failure r_1 destroys V_1, which requires 3 units of computing resources, and 1 unit of communication bandwidth with each of the two VI nodes mapped to facility nodes A_1 and A_2, respectively. Similarly, we assume that r_2 destroys V_2, which requires 5 units of computing resources, and 1 unit of communication bandwidth with each of the two VI nodes mapped to facility nodes B_1 and B_2.



(a) SOUM algorithm (b) IOCM algorithm

Fig. 4 Illustration of the SOUM and IOCM algorithm

Suppose that both facility nodes F_1 and F_2 outside G_0 are available candidate for remapping V_1 and V_2 (because e.g., they both have more than 5 units computing resources). In addition, they are connected to facility nodes A_1, A_2, B_1 and B_2 as shown, where the numbers on the links and nodes are the unit cost of using the bandwidth and computing resources respectively.

When considering r_1 first, the additional cost of mapping V_1 to F_1 is 1x3 + 4x1+3x1 =10, and that of mapping V_1 to F_2 is 1x3+7x1+(4+1)x1=15. Accordingly, mapping to F_1 and using fiber links (F_1,A_1) and (F_1,A_2) will result in a lower additional cost, than mapping V_1 to F_2, and such a mapping will be chosen by the IOCM algorithm to deal with r_1. Next, when considering failure r_2, since 3 units of computing resources and 1 unit of bandwidth resources have already been reserved on F_1 and fiber link (F_1,A_1), respectively, to handle r_2, the *additional* cost of mapping V_2 to F_1 is 1x(5-3) +5x1+(3x0+1x1)=8. However, the additional mapping cost of mapping V_2 to F_2 is 1x5+3x1+4x1=12. Accordingly, facility node F_2 and fiber links (F_1,B_1), (F_1,A_2) and (A_2,B_2) will be selected by the IOCM algorithm for r_2. *The total additional cost will be 10+8 = 18.*

If, on the other hand, the order of failures considered is r_2 and then r_1, facility node F_2 and fiber links (F_2,B_1) and (F_1,B_2) will be selected to deal with r_2. And facility node F_1 and fiber links (F_2,A_1) and (F_2, A_2) will be selected to deal with r_1. *The total additional cost will be 12+10=22.*

Note that the NSVIM algorithm may be modified to obtain the after-failure mapping for r_1 (assuming it is considered first) as follows. Initially, *MLN* should be modified to include all VI nodes except V1, and *UMLN* should only include V1. Similarly, *AFN* should initially include every facility nodes in G_0 except the node affected by r_1, while *UAFN* includes all facility nodes in G_S - G_0.

V. Simulation Result

In this section, we describe the simulation environment and present our simulation results.

A. Simulation Environment

In our experiments we use the NSFNET as our substrate network. The computing capacity at facility nodes and bandwidth capacity on the links follow a uniform distribution from 50 to 100 under unconstrained capacity and 1 to 10 when the capacity is constrained. We assume the unit computing cost (e.g., using 1000 CPU hours) to be 3 and unit bandwidth cost (e.g., using 1 Mbps) to be 1.

The VI requests are generated randomly based on four main parameters: the number of VI nodes ($|N|$), the average degree of 2 for the $|N|$ VI nodes, the average computing requirement of a VI node and the average bandwidth requirement of a VI link. The computing and bandwidth demands follow a uniform distribution from 1 to 10. For each regional failure we randomly choose two adjacent nodes to fail.

B. Performance Metrics

We use the following three performance metrics to evaluate the performance of SOUM and IOCM. The first two are applicable when the amount of computing and bandwidth resources available in the FCNS is sufficient to recover from any failure, while the last is applicable when the amount is limited.

1) Cost: This is the cost incurred by the substrate network to reserve resources to tolerate any failure scenario under unconstrained capacity. It is the sum of the computing cost on all facility nodes and the bandwidth cost on all fiber links.

2) Average number of migrations: The average number of migrations is calculated as in Eq. (3), where ntm_r is the number of VI nodes that need to be migrated to new facility nodes under failure r under unconstrained capacity.

$$TM= \quad {}_r ntm_r/|R| \qquad (3)$$

3) Recovery Blocking Probability: This is the ratio of the number of unrecoverable failure scenarios to the total number of failure scenarios.

C. Comparison of MILP, SOUM and IOCM

We solve the MILP using CPLEX 8.0 for the substrate network in fig. 1(b) with varying number of VI nodes in the VI request and compute the cost in each case. Table 1 shows that IOCM achieves the lowest cost (i.e., same cost as the MILP) and better than SOUM for a small-size problem.

Cost	Number of VI nodes in VI request		
	2	3	4
MILP	51	122	178
SOUM	57	140	190
IOCM	51	122	178

Table 1: Comparison of MILP, SOUM and IOCM

D. Comparison of SOUM and IOCM

First, we look at the case when the computing and bandwidth capacity/resources are unconstrained and compute the cost (fig. 5(a)) and average number of migrations (fig. 5(b)) of SOUM and IOCM. Then we study the influence of capacity constraints on the recovery blocking probability in fig. 5(c).

Fig. 5(a) shows the cost incurred by the initial non-survivable mapping and cost for the survivable mapping. The figure shows that SOUM incurs more cost than IOCM. IOCM has approximately 50% saving over SOUM especially when the size of VI request is small. As the number of VI nodes in the VI request increases, the gap between IOCM and SOUM reduces. The main reason is that as the size of VI requests increases the intersection/overlap between different failure solutions by SOUM increases.

Fig. 5(b) shows that the average number of migrations required by SOUM is much higher than IOCM. The main reason is that unlike IOCM, SOUM does not constrain the mapping region of the unaffected VI.

Fig. 5(c) shows that the recovery blocking probability of IOCM is higher than SOUM i.e., SOUM has better survivability when the substrate network has limited capacity. The main reason is that IOCM limits the remapping region while SOUM permits the whole VI to remap thus increasing its chances of failure recovery. Note: since FCNS has limited computing and bandwidth resources, certain failures cannot be recovered due to resource constraints. That is, even the optimal algorithm will suffer from a certain recovery blocking probability. As future work, we will investigate the feasibility of failure recovery, and determine the optimality of our heuristic algorithms when FCNS has limited amount of resources.

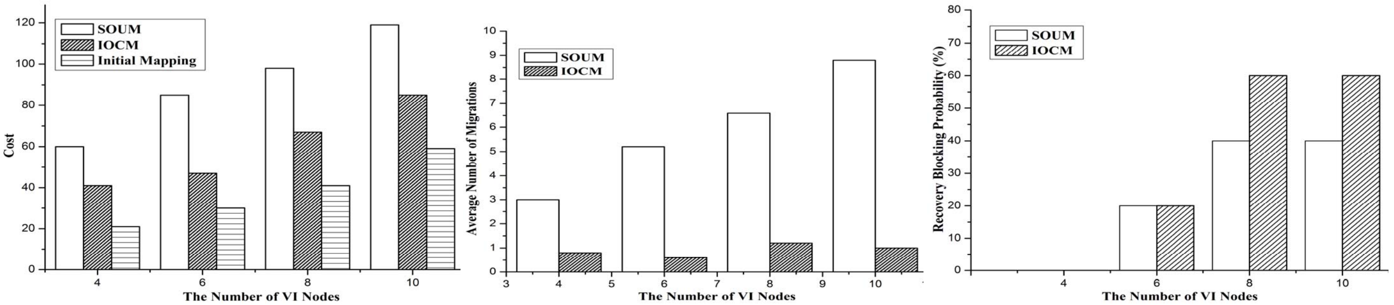
VI. CONCLUSION

Providing VI survivabilityin the presence of regional failures is critical for emerging applications using paradigms such as FCNS and Cloud Computing. In this paper, we have designed and evaluated various approaches to provisioning survivable VI which can recover from a specified list of regional failures. We have formulated the problem as an MILP and proposed a heuristic NSVIM to first solve the non-survivable VI provisioning problem. Finally,we have extended NSVIM to develop two failure-dependent survivable VI provisioning algorithms SOUM and IOCM that use separate and incremental optimization strategies. Simulation results have shown that IOCM provides more cost-efficient solution than SOUM, while SOUM has better failure recovery probability when the capacity is limited.

ACKNOWLEDGEMENT

This research was partially supported by Natural Science Foundation of China grant No.60872032 and 60972030, the National Grand Fundamental Research 973 Program of China under Grant No.2007CB307104.

The research of Dr. Qiao is supported in part by HDTRA1-09-1-0032. The information reported here does not reflect the position or policy of the federal government.



(a) Cost (b) Average number of migrations (c) Recovery blocking probability

Fig.5 SOUM vs. IOCM

REFERENCES

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared,” Grid Computing Environments Workshop, 2008.

[2] A. Baranovski et al., “Enabling distributed petascale science,” J. Phys.: Conf. Ser. 78 012020, 2007.

[3] B. Mukherjee, Optical WDM Networks. Springer, 2006.

[4] A. Sen, S. Murthy, S. Banerjee, “Region-Based Connectivity - A New Paradigm for Design of Fault-tolerant Networks”, HPSR 2010

[5] J.Luand, J.Turner, ” Efficient mapping of virtual networks onto a shared substrate,” Washington University, Tech. Report, WUCSE-2006-35, 2006.

[6] Y.Zhu, M.Ammar, “Algorithms for assigning substrate network resources to virtual network components”, IEEE INFOCOM, 2006

[7] N. M. Mosharaf at al., “Virtual Network Embedding with Coordinated Node and Link Mapping”, IEEE INFOCOM, 2009

[8] K. Lee, E. Modiano,” Cross-Layer Survivability in WDM-Based Networks”, IEEE INFOCOM, 2009

[9] S. Hwang and C. Kesselman.” A Flexible Framework for Fault Tolerance in the Grid.” Journal of Grid Computing, 2003.

[10] X.Liu, C.Qiao, and T.Wang, “Robust Application Specific and Agile Private (ASAP) Networks Withstanding Multi-layer Failures,” OFC/NFOEC, 2009, p. OWY1.

[11] Muntasir Raihan Rahman, Issam Aib and Raouf Boutaba, Survivable Virtual Network Embedding, Lecture Notes in Computer Science, 2010(4), April 2010:40-52.

[12] A. Sen, B. H. Shen, L. Zhou and B. Hao, “Fault-Tolerance in Sensor Networks: A New Evaluation Metric”, IEEE INFOCOM 2006.

978-1-4244-5638-3/10/$26.00 ©2010 IEEE