

Optimal Resource Allocation for Disaster Recovery

*Original*

Optimal Resource Allocation for Disaster Recovery / Bianco, Andrea; Giraudo, Luca; Hay, David. - STAMPA. - (2010). (IEEE GLOBECOM 2010 (Next Generation Networking and Internet Symposium) Miami, FL, USA December 2010) [10.1109/GLOCOM.2010.5683164].

*Availability:*

This version is available at: 11583/2375041 since:

*Publisher:*

IEEE

*Published*

DOI:10.1109/GLOCOM.2010.5683164

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# Optimal Resource Allocation for Disaster Recovery

Andrea Bianco\*, Luca Giraudo\*, David Hay†

\* Dipartimento di Elettronica, Politecnico di Torino, Italy, Email: {andrea.bianco,luca.giraudo}@polito.it

† Department of Electrical Engineering, Columbia University, New York, USA, Email: hdavid@ee.columbia.edu

**Abstract**—Two key elements in disaster recovery are backing up data at remote sites and reliability of services by virtualization. To support operational services and to speed up recovery process, resources should be distributed fairly and efficiently. Thus, we discuss resource allocation algorithms to support remote data storage and live virtual machines (VMs) migration. We identify two opposing forces: on the one hand, backup data should be stored as close as possible to the original site to guarantee high access speed and to minimize network load. On the other hand, upon a site failure, VM migration should minimize the impact of resuming VMs on other sites, to protect application performance and to reduce VM restoring time. We present optimal algorithms trading-off these two contrasting goals, and we compare their performance for different network topologies and resource distributions among sites.

## I. INTRODUCTION

*Off-site data protection* (copying critical data to a physically remote site) is perhaps the most fundamental strategy to ensure fault-tolerance and enable *disaster recovery*. Today, the most widely used solutions to backup data into data centers and enterprise networks rely on *Fibre Channel* (FC) [1], a networking technology designed in *Storage Area Networks* (SANs), where high access speed and reliability are guaranteed by avoiding packet losses. However, FC technology suffers from two major drawbacks: expensive infrastructure, due to relatively low production volumes, and expensive transceivers and mono-modal fibers. Furthermore, FC networks management is complex, and different from IP/Ethernet management, thus requiring additional training and increasing costs.

The iSCSI protocol [2], a SAN protocol based on TCP/IP, may become an alternative to FC. iSCSI transports SCSI commands, normally issued to directly attached disks, over TCP/IP connections, to permit access from/to remote devices which can be eventually on the Internet expanding SAN's borders out of data center's network. However, packet losses, retransmissions and large latencies may occur when dealing with IP best effort service. Thus, lower performance than those of traditional SAN architectures should be expected: this is a common drawback among similar technologies (e.g. iFCP and FCIP) which extend the SAN outside its traditional domain.

While the above described technologies enable storage virtualization giving access to virtual storage resources regardless of their physical location, application/server virtualization solutions like XEN [3] or VMware [4] are mature enough to be employed in mission critical systems to guarantee service continuity. Advanced mechanisms like *live migration* of virtual machines (VMs) are already stable and usable. This feature is especially interesting for disaster recovery applications,

because a VM can be moved transparently from site to site, accepting short idle period of few seconds, without disrupting the service.

In this paper we discuss algorithms to allocate site resources, taking into account both remote storage and VMs migration needs. We focus on a network connecting a number of sites hosting VMs and disks. Sites may be a set of data centers distributed in a geographical scenario and connected via the Internet. We emphasize that we are not addressing the classical problem of VMs placement inside data centers. Rather, we believe that increasing transmission speeds would soon enable using MAN/WAN infrastructures to provide services offered today in data centers exploiting specific interconnection architectures.

VMs run on a given site and are connected to a local disk and to a remote backup disk hosted on a different site, to provide protection in case of site failure. VMs migrate to the site hosting the remote backup in case of failures. Since we assume that local and remote data are synchronized, if an outage event occurs, the VM can be transparently restarted on the remote site without requiring the transfer of a huge amount of working data.

We define several optimization scenarios, which take into account the contradicting goals of enhancing network performance, minimizing migration costs and increasing backup service quality. Then, we present polynomial time algorithms to solve these optimization problems. Finally, we assess algorithms performance on randomly generated meshed network topologies.

## II. PROBLEM FORMULATION

The optimization goal is to distribute efficiently storage resources (*disks*) and computing resources (VMs) among network nodes (*sites*).

Let  $S$  be the set of *sites*, where a site  $s \in S$  represents a host with computing and storage resources. Sites are connected in a network through *logical links*, elements of a set  $L \subseteq S^2$ , that can be either a direct link or a multi-hop path. Links may have attributes such as *cost* or *capacity*. Virtual machines (VMs) are software applications; each VM  $v \in V$  is associated with a specific site  $s \in S$  on which it runs. Each VM running in  $s$  requires a predefined amount of storage resources at site  $s$  corresponding to a disk (or partition). To ease the disaster recovery task, each VM associated with site  $s$  requires an additional *backup* storage on site  $s' \neq s$ . Disk resources are modeled through a set of iSCSI disks  $D$ : the physical storage of each site  $s$  is partitioned in chunks named *disks*, which may

correspond either to a physical disk or to a partition. We further assume that all VMs  $v \in V$  are *uniform*, i.e., they require the same amount of storage resources (one disk or partition at the hosting site and at a remote site) and computing resources (one CPU). This is a rather strong assumption, but i) it may be very difficult to know VM's resource consumption *a priori*, ii) the approach can be rather easily generalized, and iii) dealing with non uniform VM requirements would further increase the number of parameters that should be used in the performance analysis.

With a slightly abuse of notation, we define each site  $s \in S$  as a set of VMs running on it and a set of disks residing at it. Thus,  $v \in s$  when VM  $v$  runs on site  $s \in S$ , and  $d \in s$  when  $d \in D$  resides at site  $s \in S$ . Furthermore, we define as  $S(v)$  the site where VM  $v$  is located.

The basic requirement in our resource allocation scheme is the *off-site data protection* capability: a copy of VM data must be saved in a remote site. Formal expressions of constraints are presented in Eq.(1), (2), (3) and (4). Constraint (4) is needed to drive the assignment of exactly two disk to every VM (redundancy ratio equal to 2).

$$\begin{aligned} \text{Local disk} \quad & \sum_{d \in S(v)} x_{vd} \leq 1 \quad \forall v \in V \quad (1) \\ \text{Backup disk} \quad & \sum_{d \notin S(v)} x_{vd} \leq 1 \quad \forall v \in V \quad (2) \\ \text{One service, one disk} \quad & \sum_{v \in V} x_{vd} \leq 1 \quad \forall d \in D \quad (3) \\ \text{Additional} \quad & \sum_{d \in D} x_{vd} = 2 \quad \forall v \in V \quad (4) \end{aligned}$$

The initial assignment of VMs to sites is uniform. Since all the disks are of the same size, a simple pre-computation phase assigns each VM  $v \in s$  to an arbitrary local disk  $d \in s$ , which becomes unavailable for remote backup. Let  $D'$  be the set of remaining disks after the local disk assignment.

### III. ILP FORMULATIONS AND ALGORITHMS

We investigate four problems: The *Network-Aware* problem (NAP), the *Disaster Recovery* problem (DRP), and two hybrid problems (DRC-NAP and NC-DRP). In the NAP problem the focus is on the minimization of the maximum (mean) number of (physical) hops between each VM and its backup disk, an approach to bound the end-to-end delay between VMs and backup sites to improve VMs running performance. This problem captures a major limitation of SAN protocols such as iSCSI, which suffers from large delays [5], [6]. Instead, the DRP problem deals with scenarios in which one of the sites crashes, with the goal of minimizing the overload caused by site failures on backup sites that are chosen to host VMs after migration. The hybrid problems look for solutions that jointly consider VMs running performance and recovery speed.

#### A. The Network-Aware problem (NAP)

The objective of the NAP problem is to minimize the maximum (mean) number of (physical) hops in the path between VMs and their backup disks.

We assume a given *cost* function on each logical link  $(s_1, s_2) \in L$ , denoted  $cost_{s_1 s_2}$ , which represents the number

of (physical) hops between site  $s_1$  and site  $s_2$ . The goal is to assign each VM  $v \in V$ , which reside on site  $s$ , a remote disk  $d \notin s$ , such that the maximum (mean) cost is minimized. Since backup disks must be located in a different site,  $cost_{ss} = \infty$  for each site  $s \in S$ . The ILP formulations are as follows:

Minimize maximum distance among VMs, remote disks:

$$\text{minimize } \max_{v,d} x_{vd} cost_{vd}$$

Minimize average distance among VMs, remote disks:

$$\text{minimize } \sum_{v,d} x_{vd} cost_{vd}$$

We next show that considering the average distance is equivalent to consider the maximum distance only by *scaling* the cost function as follows:

- 1) Sort the cost values in a ascending order, and let  $(c_1, c_2, \dots, c_\ell)$  be this order.
- 2) Iteratively, assign  $c'_i = c'_{i-1} \cdot |V| + c_i$ , where  $c'_1 = c_1$  and  $i = 2, 3, \dots, \ell$ .

**Theorem 1:** Let  $\bar{x} = [x_{i,j}]$  be an assignment vector that maps VMs to remote disks. If  $\bar{x}$  minimizes the sum of all costs (and therefore the average) using cost function  $c'$ , then  $\bar{x}$  minimizes the maximum cost using cost function  $c$  too.

**Proof:** Assume that  $\bar{x}$  minimizes the sum-objective function but does not minimize the max-objective function. Then, there is another assignment  $\bar{x}'$  which minimizes the max-objective function. Therefore let:

- $a = \max_{i,j} x_{ij} c_{ij}$
- $a' = \max_{i,j} x'_{ij} c'_{ij} < a$

We denote by  $b$  the index of  $a$  in the ordered cost function (that is,  $c_b = a$ ) and similarly by  $b'$  the index of  $a'$ . Since  $a' < a$ ,  $b' < b$ . Note also that, by the monotonicity of  $c'$  in the ordering of the cost function,

- $c'_b = \max_{i,j} x_{ij} c'_{ij}$
- $c'_{b'} = \max_{i,j} x'_{ij} c'_{ij}$
- $c'_{b'} < c'_b$ .

and hence,

$$\begin{aligned} \sum_{i,j} (x'_{ij} c'_{ij}) & \leq |V| \left( \max_{i,j} \{x'_{ij} c'_{ij}\} \right) \\ & = |V| c'_{b'} < c'_b \leq \sum_{i,j} (x_{ij} c'_{ij}) \end{aligned}$$

where the inequality  $|V| c'_{b'} < c'_b$  is by construction of the function  $c'$ . This contradicts the minimality of  $\bar{x}$  under the sum-objective function and the theorem follows. ■

Theorem 1 implies that to solve both sub-problems it is sufficient to devise an algorithm to minimize the mean distance between VMs and remote disks. This latter problem is exactly the ASSIGNMENT PROBLEM, and can be solved accurately and efficiently using the *Hungarian Method* [7], [8], [9]. Thus, we call HMA (Hungarian Method Average) and HMM (Hungarian Method Max) the two algorithms that solve the NAP. Time complexity is  $O(|V|^3)$  in both cases.

### B. The Disaster-Recovery Problem (DRP)

The DRP objective is to minimize the overload caused by failures on backup sites that are chosen to host VMs after migration. The problem is network-unaware: it does not consider the underlying network structure, but it is focused on the fair distribution of the VM' startup overload caused by a site crash. Thus, we often refer to DRP as a service-aware algorithm.

When a site  $s \in S$  crashes, all VMs  $v \in s$  migrate to their respective *backup site*. This *migration* process is a CPU-consuming activity, since a new VM has to be started and resources must be reserved. Therefore, the recovery process might slow down VMs already running on site.

Since it is not straightforward to profile *a priori* the amount of resources required to run a VM, a reasonable solution is to distribute uniformly among all sites the overload. Following this approach, a limited number of VMs is moved from a site to a specific backup site in case of failure, avoiding to create a large overload in some sites only.

More precisely, let  $y_{s_1 s_2}$  be the number of VMs in site  $s_1$  that use disks in site  $s_2$ . Our objective is to minimize the value  $\max_{s_1, s_2 \in S} y_{s_1, s_2}$  which is the DRP value of the assignment. The ILP formulation of DRP is as follows:

$$\begin{aligned} & \text{minimize } \max_{s_1, s_2 \in S} y_{s_1, s_2} \\ & y_{s_1 s_2} = \sum_{v \in V \cap s_1, d \in D \cap s_2} x_{vd}; \end{aligned}$$

We propose an algorithm that solves the DRP problem optimally with time complexity  $O(|S|^2|V| \log |V|)$ . The algorithm works on a graph  $G_\lambda = \langle A, E \rangle$ : the set of nodes is  $A = (S \times \{0, 1\}) \cup \{a, b\}$ , where  $a$  is a source node and  $b$  a sink node, and the set of edges  $E$  consists of three types:

- Edges from the node  $a$ :  $\{(a, (s, 0)) | s \in S\}$ . The capacity of each edge is the number of VMs in the corresponding site, namely  $\text{cap}(a, (s, 0)) = |s \cap V|$ .
- Edges within  $S \times \{0, 1\}$ :  $\{((s_1, 0), (s_2, 1)) | s_1 \neq s_2\}$ . The edge capacity is set to  $\lambda$ .
- Edges to the node  $b$ :  $\{((s, 1), b) | s \in S\}$ . The capacity of each edge is the number of disks available in the corresponding site, namely  $\text{cap}((s, 1), b) = |s \cap D'|$ .

The algorithm named DRF (Disaster Recovery Flow) iteratively constructs  $G_\lambda$  with different values of  $\lambda$  and employs an optimal *maximum flow* algorithm: let  $f_\lambda$  be the maximum flow. Notice that, since all capacities are integers,  $f_\lambda$  is integral as well. Then we have the following two lemmas related to the existence of a solution to DRP:

**Lemma 2:** If the value of  $f_\lambda$  is  $|V|$ , then there is a solution to DRP with value at most  $\lambda$ .

**Proof:** Let  $f_\lambda(e)$  be the value of  $f_\lambda$  on the edge  $e \in E$  provided by DRF algorithm and let focus the attention on edges in the central part of the graph only, excluding edges to/from nodes  $a$  and  $b$ . The objective is to extract from flows a feasible (and optimal) assignment for DRP.

Consider an arbitrary service order of the sites  $s \in S$  and for each site  $s \in S$  consider an arbitrary ordering of VMs  $V \cap s$  and disks  $D' \cap s$ .

Furthermore, consider the following assignment of remote disk to  $v_n \in s_k$ : let  $k_1$  be the minimal integer such that  $\sum_{i=1}^{k_1} f_\lambda((s_k, 0), (s_i, 1)) \geq n$  and let

$$n_1 = n - \left( \sum_{i=1}^{k_1} f_\lambda((s_k, 0), (s_i, 1)) \right).$$

Site  $s_{k_1}$  represents the first site where there are disks assigned by DRF to VMs belonging to  $s_k$  but not yet included into partial solution for DRP;  $n_1$  is a negative number and its absolute value represents the number of assignments of VMs  $\in s_k$  to disks  $\in s_{k_1}$  not yet included into partial solution.

Thus VM  $v_n$  will be assigned a remote disk in site  $s_{k_1}$ , with local index:  $n_1 + \sum_{i=1}^{k_1-1} f_\lambda((s_i, 0), (s_{k_1}, 1))$ .

It is easy to verify that assignment is legal and that its DRP value is at most  $\lambda$ , since disk availability is not exceeded due to  $G_\lambda$  structural constraints and  $y_{s_1 s_2}$  is limited by link capacity (up to  $\lambda$ ) in the central part of the graph. ■

**Lemma 3:** If the value of  $f_\lambda$  is less than  $|V|$ , then there is no solution to DRP with value at most  $\lambda$ .

**Proof:** A solution for DRP is feasible if all the common constraints described in Section II are fulfilled. This means that for every running virtual machine there must be a remote disk; since  $|V|$  is the number of virtual machines and since every VM-disk association consumes one unit of flow, the total flow must be equal to the number of VMs to have a complete assignment. Thus if the total flow is lesser than  $|V|$ , some virtual machines do not have a remote disk associated, then the solution is not acceptable. ■

The above lemmas imply that DRP is equivalent to finding the minimal  $\lambda$  for which the maximum flow of  $G_\lambda$  is equal to  $|V|$ . Notice that the value of the maximum flow  $f_\lambda$  is monotonically non-decreasing with  $\lambda$  and is bounded by  $|V|$ . Furthermore, solutions to DRP are in the range  $[1, |V|]$ , which implies that the optimal value can be found using a binary search on  $G_\lambda$  with  $\lambda \in [1, |V|]$ . Thus, at most  $\log |V|$  instances of the maximum-flow algorithm, each with complexity  $O(|S|^2|V|)$ , are needed to solve DRP: the total time complexity of DRF algorithm is  $O(|S|^2|V| \log |V|)$ .

### C. Hybrid Problems

To deal with both network performance and VM recovery service quality we define two *bi-criteria* frameworks that consider both objectives.

**DRC-NAP: Disaster Recovery Constrained-Network Aware Problem:** The objective is to choose, among all *optimal DRP* assignments, the one with minimal cost. It can be solved by the following algorithm, named DRF-MCMF:

- 1) Solve the DRP problem with the DRF algorithm. Let  $\lambda^*$  be the DRP value achieved.
- 2) Use the MIN COST MAX FLOW (MCMF) algorithm [11] on  $G_{\lambda^*}$ , where all edges from node  $a$  or to node  $b$  have cost 0, and all edges between  $(s_1, 0)$  and  $(s_2, 1)$  have cost  $\text{cost}_{s_1 s_2}$ .

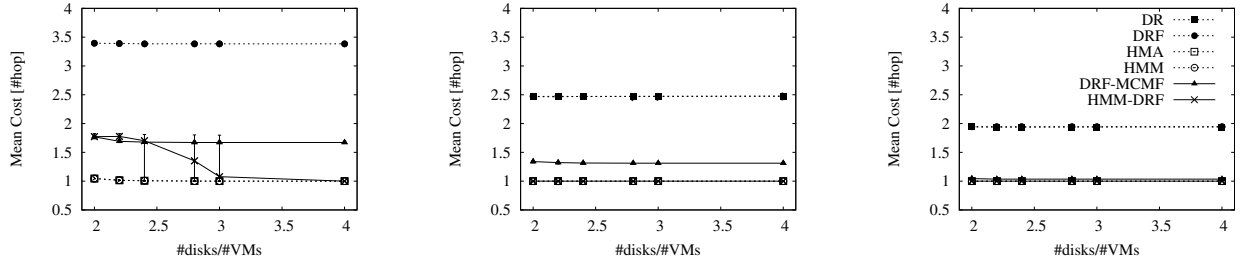


Fig. 1. Mean cost for increasing connectivity levels in mesh topologies with  $p = 0.08, 0.15$  and  $0.3$ . 1000 VMs in total.

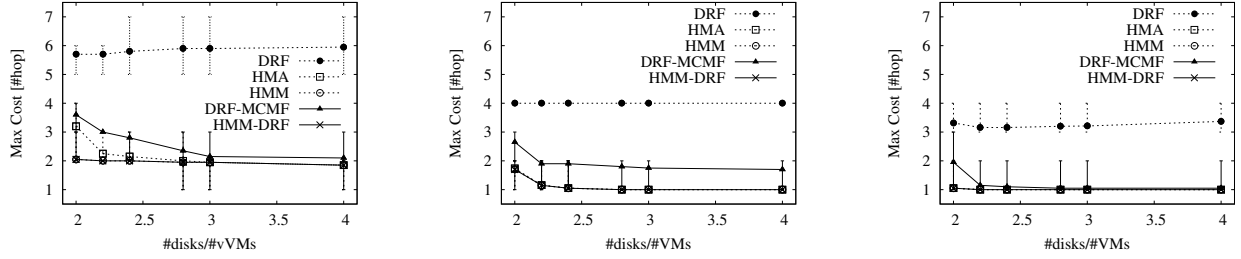


Fig. 2. Max cost while increasing connectivity:  $p = 0.08, 0.15$  and  $0.2$ , from left to right.

**NC-DRP: Network Constrained-Disaster Recovery Problem:** The objective is to choose among all *minimal cost* assignments the one with minimal DRP value. The following algorithm, named HMM-DRF, finds the optimal solution:

- 1) Solve the NAP using the HMM algorithm, ignoring the DRP objective. Let  $P$  be the maximum path length among assignments. Let  $L' \subseteq L$  the set of links with cost more than  $P$ .
- 2) Solve the DRP problem as described in Section III-B with DRF after pruning from  $G_\lambda$  of all links in  $L'$  set.

#### IV. PERFORMANCE RESULTS

We report simulation results focusing the attention on the following performance indices:

- Max and mean cost (MC and mC) of the paths between VMs and backup disks, measured in number of hops;
- Maximum number of VMs (MV) that a site must host when another site crashes.

The first two indices are related to network performance, since MC and mC indicates path length and thus end-to-end delay between VMs and disks. The MV index describes the characteristics of the algorithm from the disaster recovery point of view. Note that by minimizing the MV metric we also reduce the maximum amount of traffic between crashed site and each of its backup sites during reconfiguration phase.

We focus on random networks, whose connectivity is defined via parameter  $p$ , the probability of adding a link between two sites. Tab. I describes the considered networks.

The total number of disks available in the network ranges from a minimum value of  $2|V|$  to  $4|V|$ . Results are averaged for each network over 20 different instances, obtained by randomly assigning VMs and disks to sites. The vertical bars around the points report the minimum and maximum

TABLE I  
MAIN CHARACTERISTICS OF CONSIDERED MESH NETWORKS

$p$	nodes	diameter	min deg	max deg	avg deg
0.08	100	7	1	10	4.08
0.1	100	6	1	11	5.04
0.15	100	4	3	13	7.56
0.2	100	4	4	19	9.86
0.25	100	3	6	21	12.3
0.3	100	3	9	23	14.64

values obtained in the 20 simulation instances. Finally, curves with white symbols refer to network-aware algorithms, black symbols to network-unaware algorithms whereas hybrid algorithms are identified by solid lines and crosses or triangles.

##### A. Uniform distribution of VMs and disks

In our experiments, whose results are reported in Figs. 1, 2, 3 and 4 for a uniform distribution of VMs and disks with different randomly generated networks, we highlight four main results.

1) *Network-aware and service-aware methods:* the distinction between two classes of algorithms is evident under all metrics, since network-aware methods and service-aware methods obtain always contrasting results: HMA and HMM (network-aware) show best network metrics (mC and MC) and worse service metric (MV). On the contrary, the service-aware DRF shows worse network metrics and best service metrics. Hybrid methods find solutions able to balance the two contrasting objectives, showing, in general, performance close to those of the best methods for each specific metric.

2) *Impact of connectivity:* connectivity has a significant impact on performance results. When connectivity is high, all methods show similar performance, because there is large availability of neighboring sites and alternative paths, which

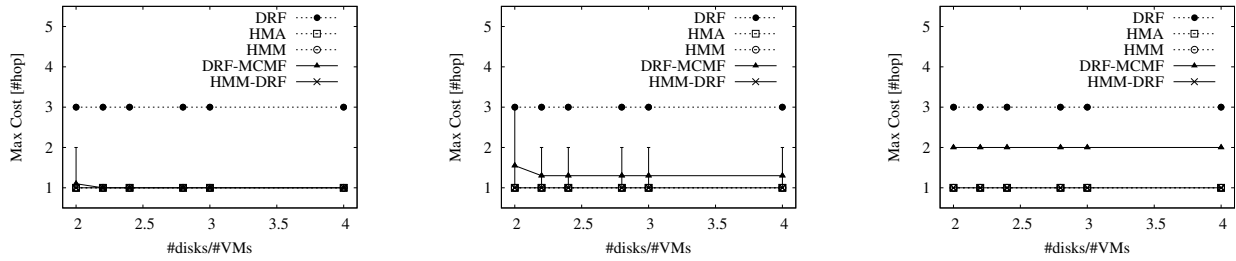


Fig. 3. Increasing number of VMs in a mesh network with  $p = 0.3$  for DRC-NAP. From left to right, 400, 500 and 1000 VMs

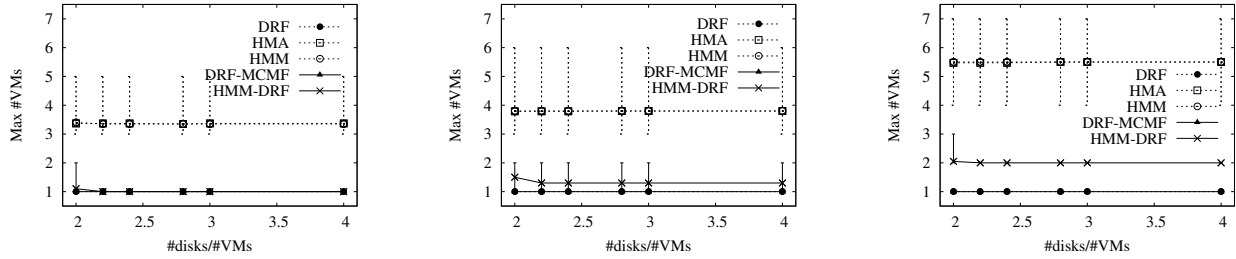


Fig. 4. Maximum number of VMs in a mesh topology with  $p = 0.3$  and 400, 500 and 1000 VMs, from left to right.

results in an efficient resource allocations, both in terms of network usage and fair distribution of VMs. On the contrary, when connectivity is limited, methods show larger differences, since the restricted availability of similar alternative solutions highlights their differences.

Indeed, this effect is evident for hybrid methods (for instance, see Figs. 1 and 2), where the first-phase algorithm is dominant. In Fig.1 HMM-DRF is better than DRF-MCMF since the network-aware method is considered first. More precisely, in the case of DRF-MCMF the DRF phase drives a fair distribution of backup disks among all (near and far) sites. The successive network-aware phase is tightly constrained by the additional distribution constraint and by the scarce availability of near sites due to limited connectivity; thus, it produces an assignment with larger mC and MC than HMM-DRF.

3) *Impact of  $|V|$  on hybrid methods:* the number of VMs has a strong influence on hybrid methods, especially in low connectivity scenarios. The impact is clear in Figs. 3 and 4. When the number of VMs increases, DRF-MCMF is worse than HMM-DRF under network metrics, but it performs better under service-metric. The explanation is similar to the previous case: the first phase algorithm drives tightly the solution produced by the second phase algorithm. The DRF phase of DRF-MCMF distributes remote disks among all (near and far) sites. This is not an issue problem if the number of VMs is limited, since many alternatives (neighbors and paths) are available to the second phase algorithm. On the contrary, when the number of VMs is larger, the number of neighbors is not sufficient anymore to permit minimum distance assignments only. Thus, also far sites are included in the solution by the network-aware phase, which shows larger mC and MC, as reported in Fig. 3. Similarly, under MV metric, HMM-DRF is

worse than DRF-MCMF, because short distance assignments force DRF to choose sites among a smaller set of near sites, thus generating larger MV as shown in Fig. 4).

4) *Hybrid methods are the best choice:* hybrid methods represent the golden choice when considering both NAP and DRP at the same time. As discussed before, these methods are able to obtain good results in all metrics as shown in Figs. 1, 2, 3 and 4. In fact they are able to reach a good compromise among network usage and fair distribution in all cases.

#### ACKNOWLEDGMENTS

These activities were developed in the framework of the FEDERICA project, funded by the European Commission. The work of D. Hay was supported by DTRA grant number HDTRA1-09-1-0057 and NSF grant number CNS-1018379.

#### REFERENCES

- [1] ANSI INCITS 373-2003, *Fibre Channel - Framing and Signaling (FC-FS)*, rev. 1.90, April 2003
- [2] IETF, *RFC 3720 (iSCSI)*, April 2004
- [3] XEN, <http://www.xen.org/>
- [4] VMware Inc., <http://www.vmware.com/>
- [5] C.M. Gauger, M. Köhn, S. Gunreben, D. Sass, S. Gil Perez, *Modeling and Performance Evaluation of iSCSI Storage Area Networks over TCP/IP-based MAN and WAN networks*, Broadnets 2005, Oct. 2005
- [6] A.Bianco, M.Modesti, F.Neri, J.M.Finochietto *Distributed Storage on Networks of Linux PCs Using the iSCSI Protocol*, HPSR 2008, May 2008
- [7] H.W.Kuhn, "The Hungarian Method for the assignment problem", *Naval Research Logistics Quarterly*, 2:83-97, 1955
- [8] H.W.Kuhn, "Variants of the Hungarian method for assignment problem", *Naval Research Logistics Quarterly*, 3: 253-258, 1956
- [9] J.Munkres, "Algorithms for the Assignment and Transportation Problems", *SIAM Journal*, 5(1):32-38, 1957
- [10] R. T. Marler, J. S. Arora, *Survey of multi-objective optimization methods for engineering*, Journal of Structural and Multidisciplinary Optimization, Volume 26, Number 6, Pages 369-395, April 2004.
- [11] R.K.Ahuja, T.L.Magnanti, J.B.Orlin, *Network flows: theory, algorithms, and applications*, ISBN:0-13-617549-X, Prentice-Hall, 1993