Network planning for disaster recovery

A.Bianco[1], J.Finochietto[2], L.Giraudo[1], M.Modesti[1], F.Neri[1]

1. Politecnico di Torino, Italy
2. Universidad Nacional de Cordoba, Argentina

IEEE LANMAN 2008
Cluj-Napoca (Romania),
September 3-6

Outline

- Introduction
- Introduction to virtualization technologies
- Considered scenario
- Problem definition and formulation
- ILP models
- Algorithms and heuristics
- Perfomance results: comparison of ILP and heuristics

Introduction

- Topic: disaster recovery
- Scenario:
  - Network of sites offering one or more “services”
  - Every site has virtualization capabilities:
    - Server virtualization
    - Storage virtualization
  - Services can migrate through sites (*VM migration*)
- Target: to define methods to assign storage resources to services in this scenario.

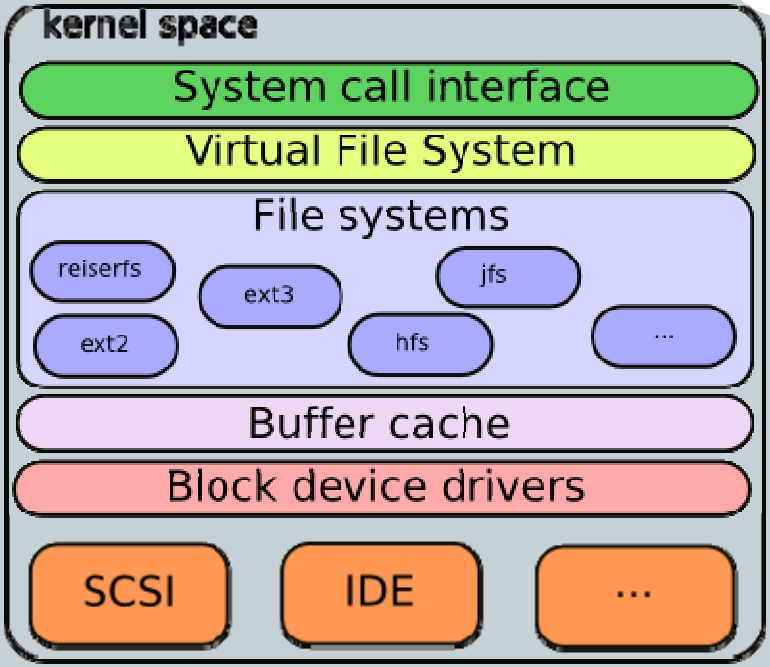
Virtualization

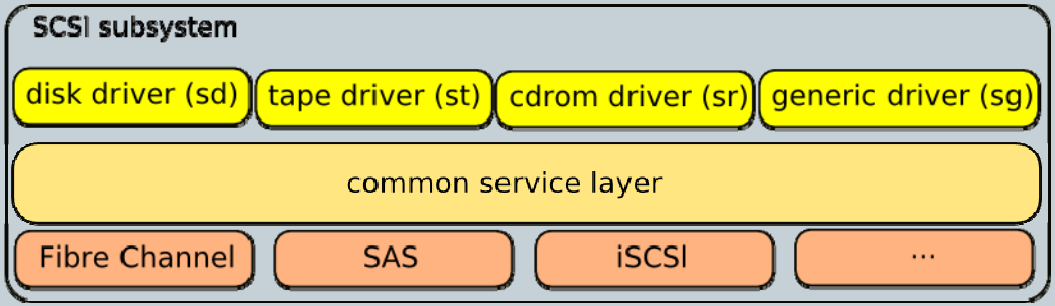
- Server virtualization
  - Partition of a physical server into isolated logical servers
  - Software available: XEN, VMware, OpenVZ, KVM, etc.
  - Interesting feature: migration
    - a virtual machine running in site A moves to another site B, without stopping the execution.
- Storage virtualization
  - Storage resources can be accessed from anywhere
  - Transparent backup of data

Storage technologies

- Local storage technologies:
  - IDE, SATA, SCSI, RAID, etc.
- Remote storage technologies:
  - File-system level:
    - NFS, CIFS (Samba)
    - FTP, SCP, rsync, etc.
    - Distributed file-systems (e.g. GoogleFS/Hadoop, IBM General Parallel File System, etc.)
  - Block level:
    - Fiber Channel (Storage Area Networks)
    - FCIP, iFCP (Fiber Channel over IP)
    - iSCSI (SCSI over IP)
    - DRBD (Distributed Replicated Block Device)

iSCSI and Linux kernel

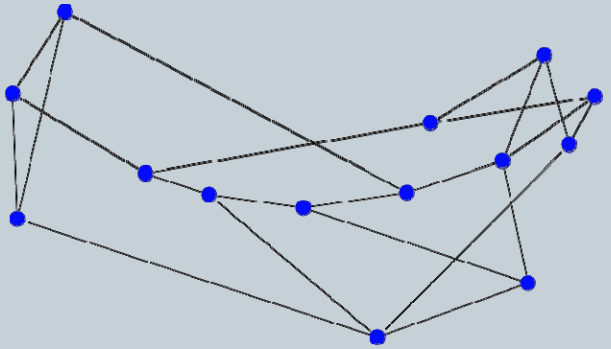


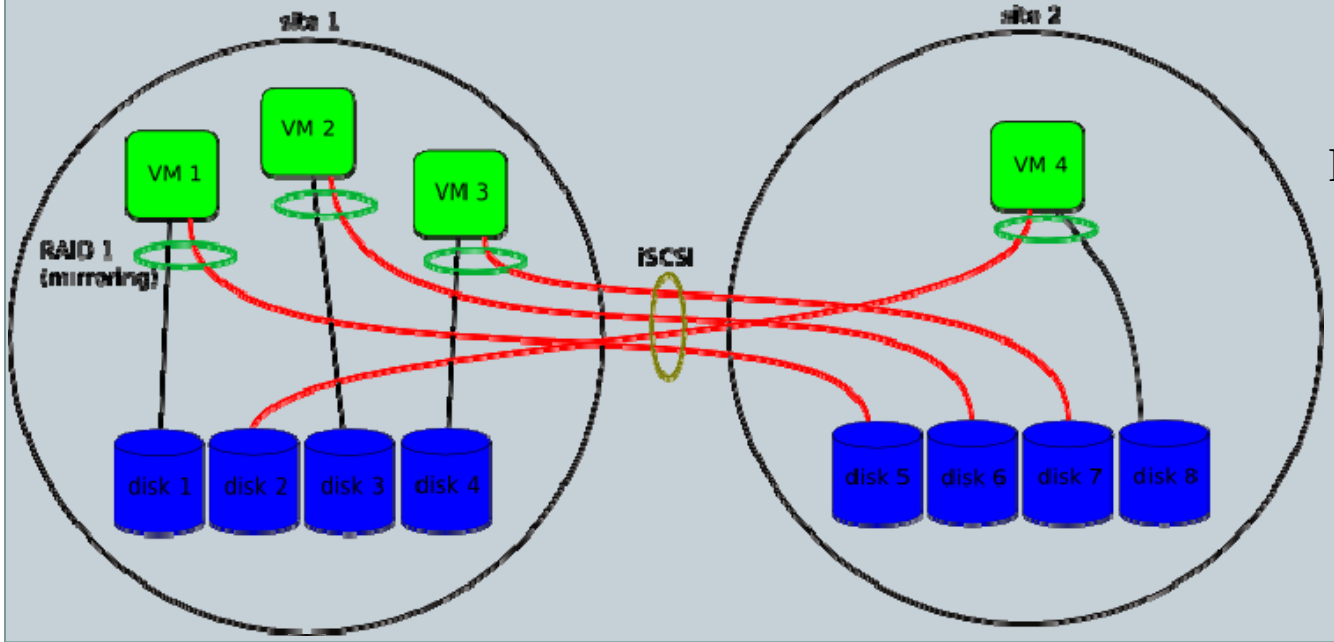


- iSCSI acts as a special driver into SCSI subsystem
  - iSCSI-disk looks like a local disk
  - iSCSI-module is stacked upon the standard TCP/IP stack

Considered scenario

Network topology
(based on NSFNET)





Relations between VMs and disks

Considered scenario, II

- Generic network architecture based on LAN/WAN technologies (TCP/IP and Ethernet)
- Different sites with virtualization enabled:
  - Different virtual machines running in every site, migration available
  - Storage resources available in every site, exported with iSCSI
- All sites belong to the same extended LAN (e.g. VPN) and to the same IP subnet (for simplicity)

Considered scenario: constraints

- Storage constraints:
  - Every running VM must have a local primary disk (on the same site).
  - Every running VM must have a secondary disk in another site.
  - Every disk must be assigned to at most one VM.
- Virtualization constraints:
  - VMs must migrate to the site hosting their backup disk

Analysis of the problem

1. Mathematical analysis:
   - 4 ILP (Integer Linear Programming) models defined and solved by CPLEX software
2. Algorithms & heuristics:
   - 4 algorithms/heuristics proposed to solve the same problems
   - Solution through custom C programs

Metrics

- Network-aware metrics:
  - Maximum bandwidth occupation registered on the most loaded link
  - Mean bandwidth occupation among active links
  - Mean cost, that is mean length of paths (number of hops) between VM and remote disk
- Virtualization-aware metrics:
  - Max. number of VMs that a site must host when another site fails (migration process)

Problem formulation: notation

- S set of sites
- V set of virtual machines
- D set of disks
- v index of the set of VMs
- d index of the set of disks
- s(v) site where the v-th VM is running
- $x_{v,d}$ binary variable, 1 if the v-th VM is associated with the d-th disk, 0 otherwise

Problem formulation: basic constraints

One local disk per VM

$$\sum_{d \in s(v)} x_{v,d} = 1, \forall v \in V$$

One remote disk per VM

$$\sum_{d \notin s(v)} x_{v,d} = 1, \forall v \in V$$

One VM per disk

$$\sum_{v} x_{v,d} \leq 1, \forall d \in D$$

First version of constraints

Problem formulation: basic constraints

One local disk per VM

$$\sum_{d \in s(v)} x_{v,d} \geq 1, \forall v \in V$$

One remote disk per VM

$$\sum_{d \notin s(v)} x_{v,d} \geq 1, \forall v \in V$$

One VM per disk

$$\sum_{v} x_{v,d} \leq 1, \forall d \in D$$

Two disks per VM

$$\sum_{d} x_{v,d} = 2, \forall v \in V$$

Final version of the constraints, used to speed up the solution process

ILP model 1: bandwidth

- iSCSI has poor throughput with large delays ([1], [2])
- Objective: control the congestion and limit (indirectly) the maximum delay

Objective Function

$$\min \max_{h,k} \sum_{v,d} B_v \times y_{vdhk} \times x_{vd}$$

- B_v bandwidth request of v-th VM
- y_{vdhk} binary variable, 1 *iff* the link between sites s_h and s_k is used by the flow from v-th VM to d-th disk, 0 otherwise

ILP model 2: hop count

- Alternative approach to limit delays
- Objective: bound end-to-end delay of flows limiting the average number of hops

Objective Function

$$\min \sum_{v,d} h_{vd} \times x_{vd}$$

- h_{vd} number of hops between v-th VM and d-th disk (determined by Dijkstra routing)

ILP model 3: service

- The method is not network-aware
- Migration is a CPU-consuming activity
- Objective: limit the CPU-overload due to migration, without considering network metrics

Objective Function

$$\min \max_{h,k \in S} N_{hk}$$

$$N_{hk} = \sum_{v \text{ in } s_h, d \text{ in } s_k} x_{vd}$$

- N_{hk} number of VM hosted in site h-th having remote disk in site k-th, that is the number of new VMs that the site k-th has to restart in case of failure of the site h-th

ILP model 4: constrained service

- Combined model: firstly ILP model 2 (hop count) to obtain the maximum number of hops and then ILP model 3 (service) with the additional constraint:

$$x_{vd} \times h_{vd} \leq \text{max\_hop} \quad \forall v, d$$

When VM v-th and disk d-th are not associated, this constraint always holds since:

$$x_{vd} = 0$$

$$\text{max\_hop} \geq 0$$

Algorithms & heuristics

- We propose some algorithms & heuristic to solve the problems 1-3
  - Targets:
    - solving the same problems in a more efficient way
    - obtaining an acceptable approximation
  - Results:
    - ILP model 2 solved exactly by MWA algorithm (Hungarian method)
    - ILP model 1 and 3 solved approximately by heuristics LPT and DR
    - MSA proposed as a reference method, no relations with ILPs

Alg. & heur. 1: LPT

- LPT = Longest Processing Time
- Idea coming from CPU schedulers: serve jobs in decreasing order of duration (largest job first)
- In our case, assign first a disk to the VM with the largest bandwidth request, using the least occupied path.
  - Focus on network parameters: bandwidth
  - Greedy approach: not always able to find a feasible solution (depending on the distribution among sites of available disks)
  - Proposed to approximate ILP model 1

Alg. & heur. 2: MWA

- MWA = Minimum Weight Assignment
- Consider the limitation of iSCSI: minimize the length of paths (in number of hops) between VM and disk
  - Focus on network parameters: path length
  - Solution: bipartite graph, Hungarian algorithm (Minimum Weight Assignment)
  - Proposed to solve exactly ILP model 2

Alg. & heur. 3: DR

- DR = Disaster Recovery
- Consider the migration process: limit the CPU overload in case of failures
  - Focus on migration process: minimize the maximum number of new VM to restart (in a site) in case of failure
  - Solution: every site selects uniformly its backup disks among all the sites.
  - Proposed to approximate ILP model 3

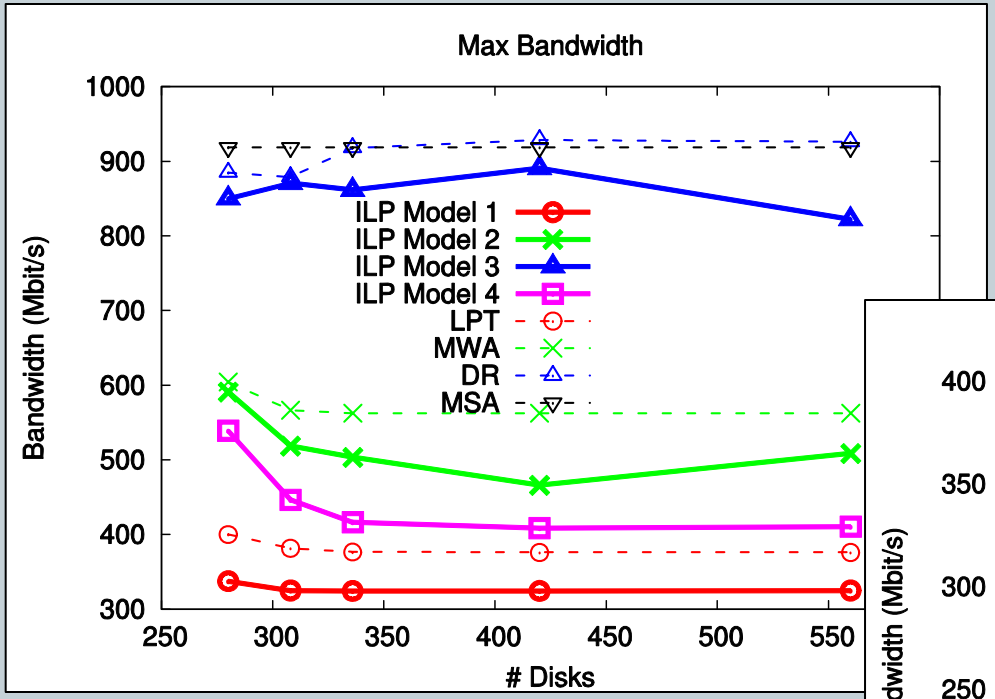
Alg. & heur. 4: MSA

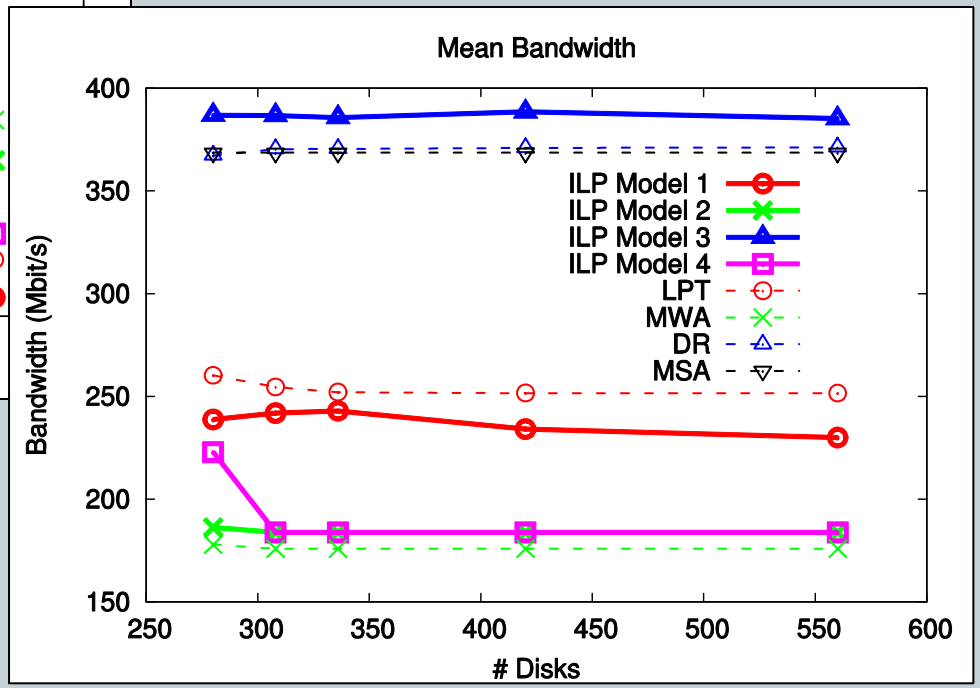
- MSA = Maximum Size Assignment
- Maximize the assignment, network parameters not taken into account
  - Focus on complete assignment: every VM must have a backup disk
  - Solution: bipartite graph, max flow algorithm (Ford-Fulkerson)
  - No counterpart in ILP models, proposed as a simplified version of the previous heuristic (DR)

Simulation parameters

- Results based on a NSFNET-like topology
  - 14 sites, 22 links (infinite capacity), max node degree = 4
  - 140 virtual machines distributed uniformly among sites, number of disks from 280 (2*VM) to 560 (4*VM)
  - VMs have a bandwidth request distributed uniformly between 10 and 100 Mbit/s
  - All sites belong to the same extended LAN (for simplicity, to avoid IP configuration & routing issues in migration process)
  - Results averaged on 20 network instances (different distribution of VMs and disks)

Results: bandwidth

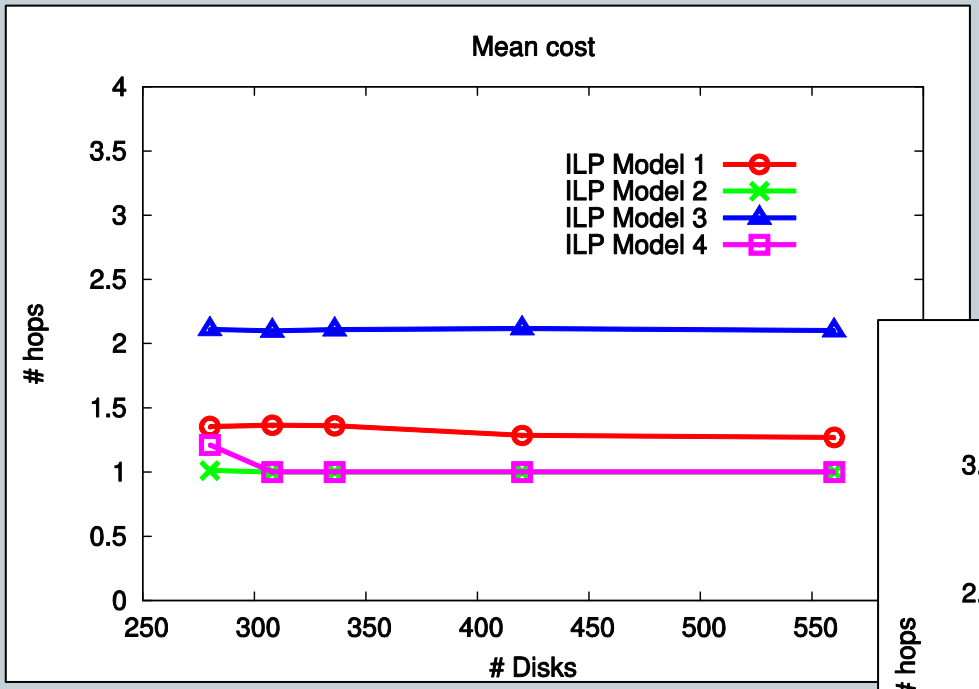


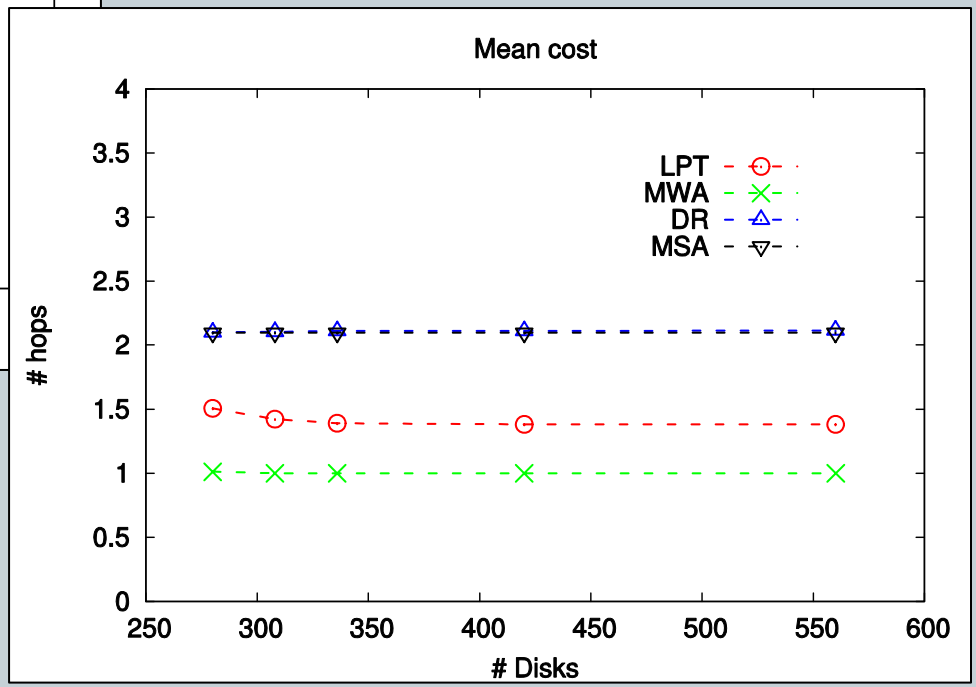


Results: bandwidth, II

- Heuristics approximate well ILP solutions
- ILP 3 - DR are the most bandwidth-hungry, since they do not consider network metrics
- ILP 4 (= 2+3) is influenced mainly by ILP 2, obtaining lower bandwidth metrics than ILP 3

Results: mean cost

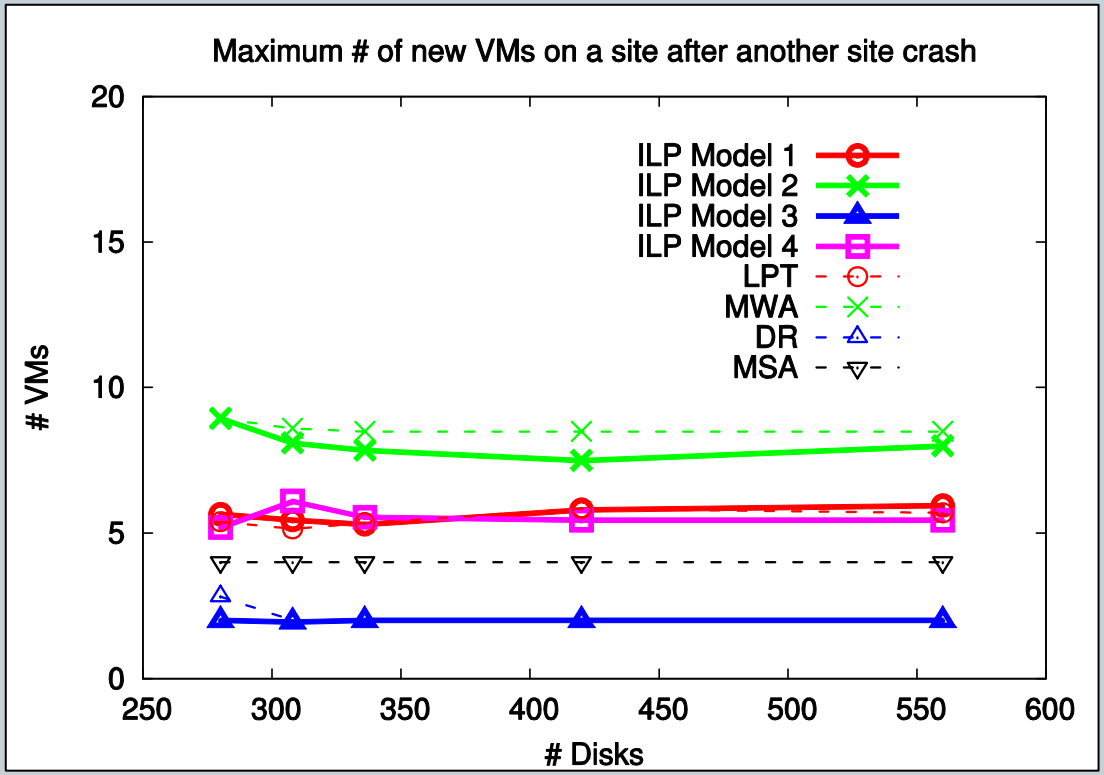




Results: mean cost, II

- Heuristics approximate well ILP solutions
- Network-aware methods select backup disks from immediate neighbours (lower hop count and lower bandwidth utilization)
- Service-aware method (ILP 3) selects disks in a wider area (higher hop count and bandwidth utilization)

Results: max number of VMs



Results: max number of VMs, II

- Heuristics approximate well ILP solutions
- Network-aware methods create large virtualization overhead in case of a failure
- ILP 3 – DR distributes uniformly the overload among all the sites

Conclusions

- Well-known algorithms can be applied to disaster recovery topic
- Trade-off between network and virtualization metrics:
  - lower bandwidth <-> higher overload
  - higher bandwidth <-> lower overload
- ILP 4 is a promising intermediate approach, obtaining good results in both metrics

Bibliography

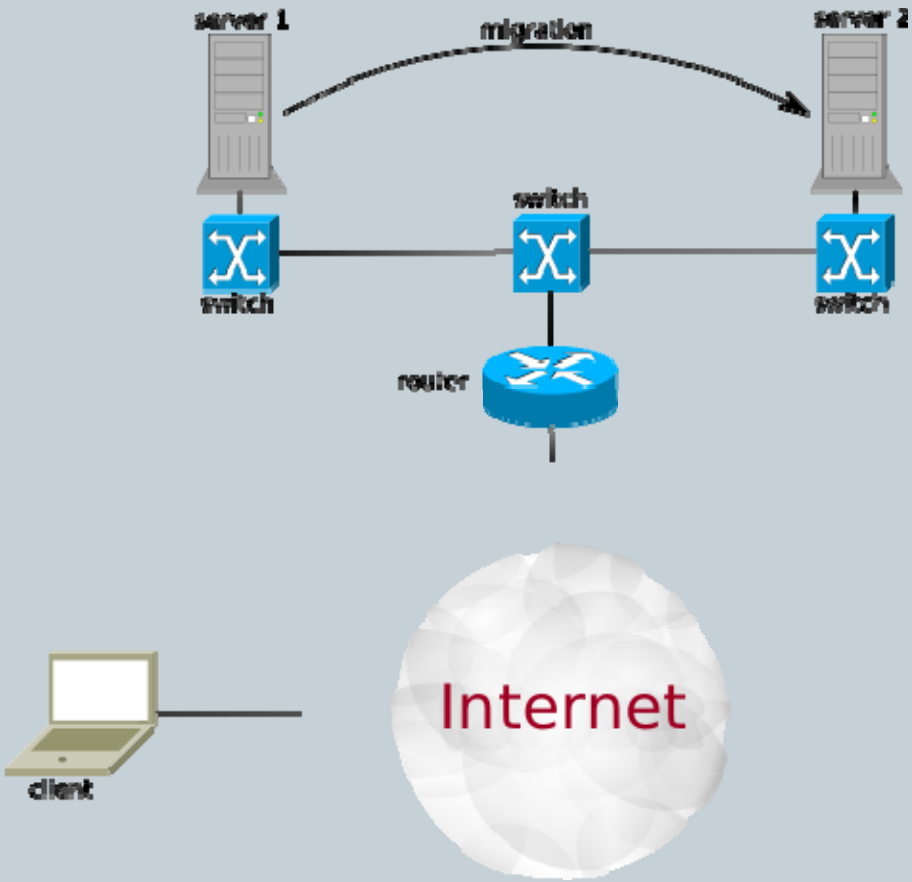
1. C.M. Gauger, M. Köhn, S. Gunreben, D. Sass, S. Gil Perez, *Modeling and performance evaluation of iSCSI Storage Area Networks over TCP/IP-based MAN and WAN networks*, Proceedings Broadnets 2005, Oct. 2005
2. A. Bianco, M. Modesti, F. Neri, J.M. Finochietto, *Distributed Storage on Networks of Linux PCs using the iSCSI protocol*, Proceedings HPSR 2008, May 2008

Inutilizzate

Features of server virtualization

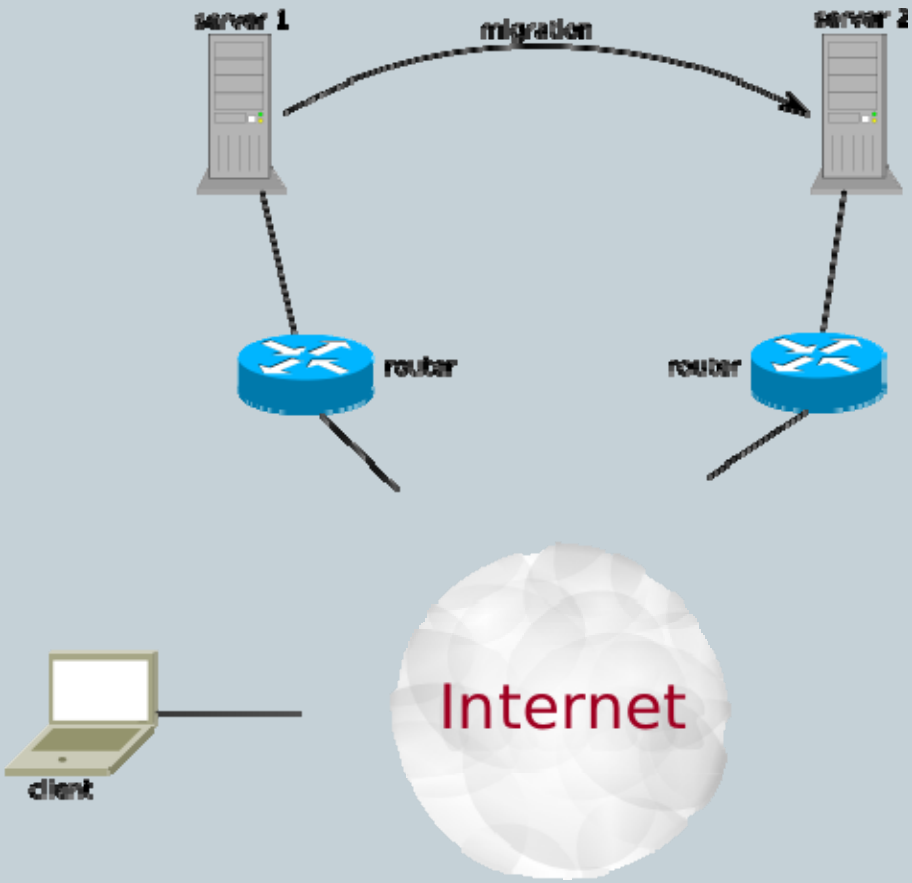
- Migration
  - a virtual machine running in site A moves to another site B, without stopping the execution.
  - Images of virtual machines are stored in every site, no need to transfer data during the migration process (if an external storage resource is used to store *running* data)

Migration examples: case 1



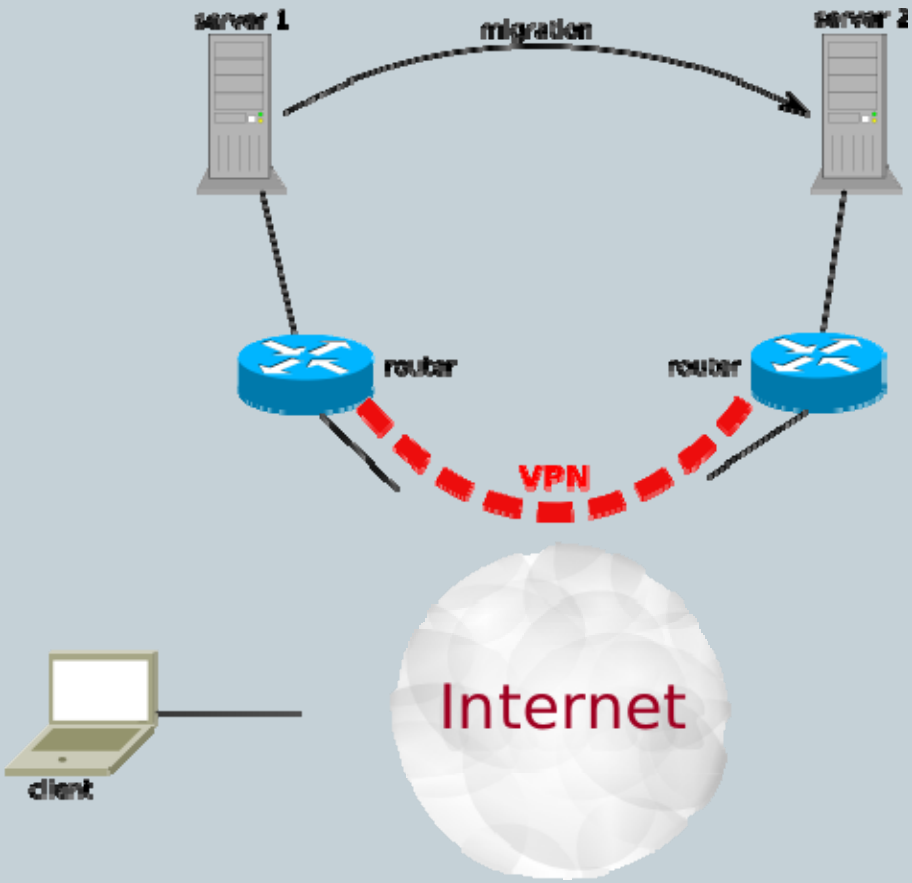
- First case: application/OS –image migrates between two nodes on the same LAN
  - No change of IP address is needed
  - The client does not notice the difference

Migration examples: case 2



- Second case: the application/OS – image migrates between two nodes on different LANs
  - Change of IP address is needed
  - The client must communicate with a different server (higher level management scheme is needed to update server IP address)

Migration examples: case 3



- Third case: the application/OS – image migrates between two nodes on different sites, belonging to the same extended LAN (e.g VPN, *Virtual Private Network*)
  - Change of IP address is not needed
  - The client must reach a different site, but the routing is done by VPN routers

Storage virtualization

- **Introdurre meglio questo concetto**
- Remote disks residing on another site mounted locally
  - SAN: a specialized network (principally Fiber Channel) substitutes the internal cables between processing units and storage resources with a packet-switched network
  - LAN/WAN: specialized protocols to transport SCSI commands over TCP/IP protocol suite (iSCSI, iFCP, FCIP)