Multi-Objective Optimization for Virtual Machine Migration on LANs for Opportunistic Grid Infrastructures

Nathalia Garcés, Nicolás Ortiz, David Mendez, Yezid Donoso

Departamento de Ingeniería de Sistemas y Computación
Universidad de los Andes
Bogotá, Colombia
{n.garces26, dg.mendez67, n.ortiz980, ydonoso}@uniandes.edu.co

Abstract—**This paper illustrates how to apply a solution for a multiple objective problem in a simple and efficient way through the case study of an example where we must copy a single file, in this case a virtual machine, on to all computers of a LAN. Our solution is intended to be used for the creation of Virtual Clusters, which are clusters composed of virtual machines that execute on opportunistic grid infrastructures. We specify the restrictions through a mathematical model and then proceed to implement a two-part solution: First, we use the solver CONOPT to determine the Pareto frontier; then we implement an evolutionary algorithm to generate possible solutions, and match them to the Pareto frontier. Finally, we evaluate our solution as an efficient way of solving the problem through the result's attributes and conclude which are the advantages of using evolutionary algorithms to find an answer for a multiple objective problem (time, possible solutions and variability between them).**

Keywords- MOP; SPEA; resource sharing; Grid Computing.

I. INTRODUCTION

Often, we find that there are problems for which no unique answer is clear and that there are many factors to consider before a decision is made. Take for example an archive copying problem where we have to give a single file to many computers connected to a Switch-based LAN network in the least amount of time. But, if we also say that we have to do it while other people are using those computers, so we cannot occupy the entire span of the bandwidth, then it becomes a bit more difficult to nail down an answer. This type of problem is called a multiple objective problem.

This problem is found on opportunistic infrastructures, where the goal is to take advantage of the unused capabilities of desktop computers. On a university campus, there are computer labs in which students can develop their daily activities. These daily activities do not fully use the processing capability of the computer. By the use of virtual machines, it is possible to create Virtual Clusters (VCs) running on desktop computers to be used as a part of a grid infrastructure. This is the goal of the infrastructure UnaGrid [1], which allows the creation of VCs and their execution on desktop computer labs at the Universidad de Los Andes. However, the Virtual Machines (VM) must be copied to each desktop computer.

In order to deploy a new cluster it is needed to transfer the VM from a source computer to the rest. This can be seen as a file transfer from one node to every other node in a network. Because the transfer is on an opportunistic infrastructure, it must be designed to not disturb the users of the computer lab.

In this paper, this case study is analyzed. The article begins by addressing other solutions for similar problems and then describes the specifics of this problem. After that, it evaluates the many components of its answer, including the SPEA-based algorithm that is used (its chromosome design, its population selection process, its genetic operators and its exit point). Finally, the results to our solution are explained and it concludes by specifying how it pertains to multiple objective problems.

II. RELATED WORK

Efficient data management has been always a challenge on large scale infrastructures. On Computational Grids, it is required to have a flexible, fast, reliable, and secure way of sharing resources across sites. Several solutions have been developed while looking for an efficient way of file sharing between numerous nodes.

On Grid environments, Grid FTP [2] provides efficient and reliable data transfer between computing nodes located among different sites. Some transfer services have been built on top of GridFTP and added to the Globus Toolkit in order to provide fault recovery mechanisms. Nevertheless, GridFTP is designed to transfer large amounts of data across different networks. Another work based on FTP, designed for parallel transfer sessions is P-FTP. This protocol is also intended to be used across different networks. Since our work aims to analyze the transfer inside a single LAN, it requires a different approach.

On Data Grids, several efforts have been made to manage the data transfer between nodes [3] [4] [5]. On these infrastructures, the data is distributed on replicas among them. There are applications that analyze the bandwidth status between connections, in order to adjust the workloads and to reduce the file transfer time. They try to adapt the file transfer to network links which do not have a predictable or

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-174-8

stable bandwidth. These algorithms also seek to reduce idle time wasted waiting for the slowest server. This is the kind of approach that is useful in this work's solution. However, our context does not have replicas for file transfer; the file must be transferred from one node to every other node on a LAN.

In other words, it is needed to transfer one virtual machine from one node to every other node on a LAN of interconnected physical desktop computers. The solution proposed by the Ohio State University [6] migrates virtual machines by using RDMA (Remote Direct Memory Access), which allows a computer to access the main memory of another without involving the operating system. This schema permits a very high-throughput data transfer between the nodes; it also reduces the transmission overhead up to 80%.

Furthermore, this solution requires a special configuration of each computer involved on the cluster. In the UnaGrid infrastructure the VMs that constitute the VCs are deployed on common computer labs, in which the computers are not controlled by the administrators of UnaGrid. This restriction makes it unfeasible to conFigure the physical machines to support RDMA.

On the other hand, multicast can save great amounts of bandwidth if it is used for file transfer in a LAN. Earlier, UnaGrid had been extended to copy VMs using reliable multicast. However, the firewall configuration of the computer labs blocks any multicast transfer when it involves large files. Generally, the VMs copied on to the computer labs consist of files greater than 5 GB. So another approach is needed, rather than multicast based schemas.

Some solutions involve the use of multicast-based schemas such as overlay multicast [7]. That particular solution describes a method for reliable data transfer based on this protocol, achieved by the usage of the application layer. They create a binary distribution tree, in which each node acts both as a sender and a receiver of packets using TCP, and it changes its structure according to the network condition. This solution is not in the scope of this paper but will be explored as future work.

III. PROBLEM STATEMENT

The situation starts off with a topology where *n* terminals are connected to a switch with *Cs* capacity and all of the connections are Symmetric DSL with the same amount of Bandwidth. (For practical purposes, assume that *Cs* is greater than or equal to the sum of the connections of the terminals, establishing that the network will not collapse if all of the terminals are using their connection to their full extent.)

In one of these terminals, there is a large archive (a VM) that requires to be transfered to other computers throughout the LAN. UDP cannot be used because of firewalls and it must be done without the users of the computers knowing about it, so the Switch's capacity cannot be consumed too much or else the users will start to notice a lag in their connection.

The objective is to be able to copy the large archive onto all of the computers in the least amount of time and having all of the computers finish downloading the archive at approximately the same time.

A. Mathematical Formulation

1) Objective Functions

a) Min (Bwmax – Bwmin)

Minimize the waste. This means we want all the nodes to end at approximately the same time.

b) Min (Tf)

Minimize the time that it takes to copy the information to the last node.

2) Constraints

a) $C_{ij} - U_{ij} = R_{ij} \quad , \quad \forall i,j \in E$

The remnant R_{ij} that can be used, from the connection of the ith terminal, to transmit the file is equal to the capacity C_{ij} of the connection minus the capacity U_{ij} being consumed by other functions or the user.

b) $\alpha_{ij} = \%R_{ij}$

The percentage of use α_{ij} is equal to a defined percentage that will be used from the remnant of R_{ij}. Note that it is a percentage, not a capacity.

c) $Bw_{min} = min(\alpha_{ij}R_{ij})_{j \in E} \quad , \quad i = 0$ *(download)*

The minimum bandwidth of the configuration is equal to the minimum actual usage of the connection, which is the percentage of use α_{ij} times the remnant of the connection R_{ij}. Note that the usage is only from the central switch ($i = 0$) to a node.

d) $Bw_{max} = max(\alpha_{ij}R_{ij})_{j \in E} \quad , \quad i = 0$ *(download)*

The maximum bandwidth of the configuration is equal to the maximum actual usage of the connection, which is the percentage of use α_{ij} times the remnant of the connection R_{ij}. Note that the usage is only from the central switch ($i = 0$) to a node.

e) $T_f = \frac{A}{Bw_{min}} + t * (n-1)$

The time it takes for the final node to finish downloading the file is equal to the size of the file A divided by the minimum Bandwidth Bw_{min}, plus the time it takes to establish the connection and start sending the information to the next node (t) times the number of nodes n minus one.

f) $\sum_{ij \in E} \alpha_{ij}R_{ij} \leq k * Cs$

The sum of the actual usage of the connections must be less than or equal to the Switch's capacity Cs times a defined percentage k.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-174-8

g) $0.1 \leq \alpha_{ij} \leq 0.9 \quad , \quad i = 0$

The percentage of use α_{ij} must be less than or equal to 0.9, but more than or equal to 0.1 in order for the program to work.

IV. NETWORK

The network is a star graph due to the fact that none of the terminals has a direct connection between them, but may communicate to each other through a central switch. In this particular problem a specific configuration has been determined for the simulation:

- A = 4 GB
  The size of the file that is to be transmitted is equal to 4 GB.
- Cs = 500 Mb/s
  The capacity of the Switch, which is the total amount of bandwidth (being used at the same time) it can sustain without running into problems, is 500 Mb/s.
- k = 0.5
  In this case, half of the total capacity of the switch is used. Thus, k is the percentage restrain over the capacity of the switch usable by the solution.
- t = 1 s
  Assume that the delay of establishing a connection and to start sending the file between terminals is 1 second long.
- n = 8
  The total nodes in the network will be eight. Remember than in all of the cases, one of them already starts with the file.

V. SOLUTION

Once the topology to test the MOP has been established, two methods have been selected to find the best solutions. One approach is to convert the MOP in a single objective problem and the other is to optimize both of the objective functions simultaneously.

For the mathematical problem the first approach was selected: a classic method called weighted sum. This method would provide the real Pareto frontier of the problem, or at least some of the solutions. Since the problem was not established as convex or not, it can't be guaranteed that this method would be the best approach to discover all the solutions of the Pareto frontier.

Therefore, the second approach was necessary: a meta-heuristic method called Evolutionary Algorithms (EAs). EAs are not entirely probabilistic because they have some intelligence that they use to find the correct solutions; also their computational time is lower than a probabilistic method. In addition this method breaks the convexity problem, which means that it can be used to find the entire Pareto frontier.

Multi objective evolutionary algorithms have a lot of implementations, but SPEA algorithm [8] was deemed appropriate enough for the problem.

VI. WEIGHTED SUM METHOD

To find the solution of the mathematical model the classic method was implemented with the solver CONOPT. Since this approach for MOP uses only one objective function (F), the weights assigned to each sub-function (f1, f2) were varied by increasing or decreasing 0.1.

$$F = w1 * f1 + w2 * f2$$

When executing the solver the first objective function was always converging to zero. It was then necessary to impose a new restriction to the mathematical model so that the value could vary.

$$Bw_{min} * 0.1 \leq (Bw_{max} - Bw_{min})$$

In other words, this restriction means that the value of F1 must be at least ten percent of Bwmin.

This variation of the model resolved the difficulty. At last, the solver was executed ten times and yielded 8 points of the Pareto frontier. These points are the red ones shown in Figure 4.

VII. SPEA

The solutions granted by this method are the blue ones shown in Figure 4.

The overall algorithm is shown in Figure 1. Next, the implementation of each step of the algorithm is described.

Begin
Generate initial population *P* randomly. Make *P* feasible according to constrains
Generation $G = 1$
WHILE $G < G_{max}$
Calculate objective values for every solution in *P*
Find non-dominated solutions in *P*
Incorporate non-dominated solutions of *P* to *P'*
IF number of solutions in *P'* exceeds maximum number allowed
Purge *P'* using *clustering*
End IF
Calculate *fitness* for every solution in *P* and *P'*
Apply *selection* to *P* + *P'*
Apply *cross over* and *mutation* to generate new population *P*
Make *P* feasible according to constrains
$G = G + 1$
End WHILE
Print results in output file
End

Figure 1. SPEA algorithm structure.

A. Individual (chromosome)

A chromosome represents a solution to an EA problem. This solution is represented as a vector of genes in which each one contains a node of the topology followed by the alpha and it's respective Bw.

Node 1 $\|\alpha_{0,1}\|Bw_{0,1}$	…	Node i $\|\alpha_{0,i}\|Bw_{0,i}$	…	Node 8 $\|\alpha_{0,8}\|Bw_{0,8}$

Figure 2. Coding of the chromosome.

B. Initial population

A population is an array of chromosomes. As stated in Figure 1, this initial population (P) is generated randomly. In this specific implementation it starts at 100 chromosomes

with random alpha values. In addition it's guaranteed that each chromosome is a feasible solution, which means that it takes into account every constraint.

C. Fitness calculation

The higher the fitness of a chromosome, the better the solution is. This means that the values for the non-dominated chromosomes are higher than the dominated ones. To achieve this each chromosome is compared with the entire population P, and for every chromosome that it dominated we added one to its fitness value. At the end of the comparisons the chromosomes with higher fitness were deemed the best solutions so far, so the 10% of the population with the highest fitness where moved to the elitist population (P').

D. Clustering

Although clustering is a very important process to maintain P' small, it wasn't implemented. Instead, every 150 generations, 60% of the chromosomes with the lowest fitness were deleted.

E. Genetic Operators

1) Selection

It was unknown if the values of the fitness of the chromosomes were very far apart. Which is why the Ranking selection method was applied. This way each of the chromosomes had a guaranteed chance to be chosen to take part in the combinatorial process.

2) Crossover

For this operation to take place a 70% possibility was determined.

For its implementation the simple crossover, with two parents/chromosomes involved, was chosen. A number between 1 and 7 was picked at random and the result lead us to the crossover point.

At the end of this process the offspring was verified as a feasible solution, if it wasn't it was discarded.

3) Mutation

For this operation to take place a 30% chance was stipulated.

Also, two types for this algorithm were implemented, each one with 50% chance. The first one is a permutation and the second one is changing the value of a random alpha value. This operation uses one parent and produces another one. This new chromosome was also checked as a feasible solution and discarded if it wasn't.

4) Exit point

It was concluded that 4000 generations where enough for the chromosomes to converge into optimal solutions.

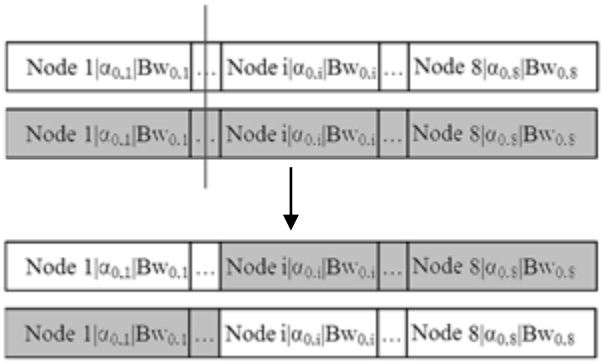

Figure 3. Crossover Operation.

VIII. RESULTS

A. Pareto Optimality

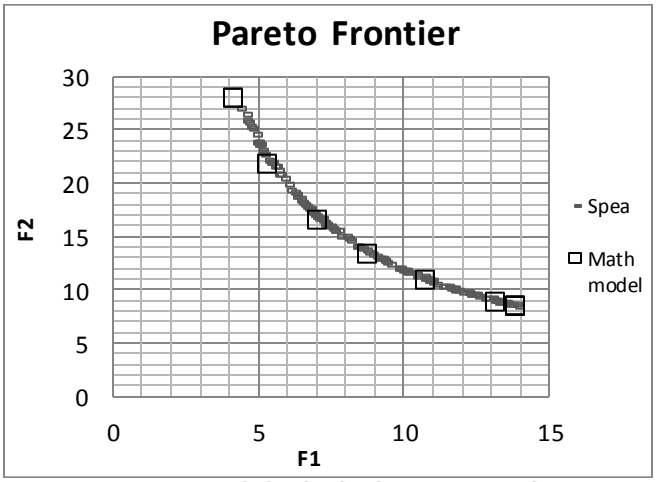

Figure 4. Pareto Frontier calculated with solver CONOPT and SPEA.

B. Evaluation

The solution is evaluated by providing 6 metrics [7]: the first three metrics indicate quantity and the last three indicate quality. It is important to consider more than one metric because only one doesn't take into account all of the performance of our SPEA algorithm.

1) GVND

The non-dominated points that present at the end of the SPEA execution are 135.

2) RGVND

$$\frac{135}{8} = 16{,}875$$

Where 8 is the number of the points found by the mathematical solver.

3) GRVND

The sum of the points found by SPEA and by the math model is 143.

4) Error

$$\frac{135 - 143}{135} = -0{,}059$$

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-174-8

5) Generational Distance

By applying the formula, the value of this metric is 0,102.

6) Spacing

$$s = 0{,}12$$

The ideal value of the last three metrics is zero, however our values are not that far apart. This means that we have a good behavior of our algorithm. In other word, the distribution of our solution is adequate, the distance between our solutions and the mathematical model is minimal and our error (our solution compared with the solution given by the solver CONOPT) is nearly insignificant.

In addition, the first three metrics indicate that we have a considerable amount of solutions, which is something good.

IX. CONCLUSION

Our model can find an efficient way of transferring a file from one source node to every other node on a LAN. This transfer is done in an opportunistic way, it is intended to adapt to the use-percentage of each link of the network. So our algorithm can find a solution in which the user does not perceive any change on the quality of service of the network due to the transfer.

From both our solutions calculated from the evolutionary algorithm and CONOPT we obtain practically the same Pareto Frontier. This can be proven by our result for generational distance. This tells us that our evolutionary algorithm is not obtaining only local minimums, and that our mathematical model is well defined for our problem.

We obtain more diversity of solutions from our evolutionary algorithm. This shows that our evolutionary algorithm can provide more information in case we wanted to develop software to calculate the best way of migrating virtual machines on a LAN.

X. FUTURE WORK

Our algorithm has proven a good performance and efficiency in small networks. However it is necessarily to test it in bigger networks because in these tests is where it can evaluated the scalability of the proposed solution and the complexity of the algorithm.

Also in our algorithm, we simplified the way in which we reduce the size of the elitist population. The SPEA algorithm proposes that to control the size of the elitist population some solutions have to be removed by using Clustering [8]. We simplified this calculation by using a different algorithm to reduce the population. When our elitist population reaches a size S greater than a defined maximum M, what we do is to calculate the fitness inside the elitist population and then we eliminate the last M-S solutions. We suggest as a future work to implement clustering for reducing the size of the elitist population.

In our solution we propose a way of transferring the file by using a pipeline. A future work could consider a different schema, for example binary trees, or n-ary trees where the root is the node that is the source of the file, and the tree defines how to organize the transfer. It must be considered that these kinds of solutions require a different mathematical model.

REFERENCES

[1] H. Castro, E. Rosales, M. Villamizar and A. Jiménez, "UnaGrid: On Demand Opportunistic Desktop Grid" 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (VIC 2010), May. 2010, pp. 661 - 666, doi: 10.1109/CCGRID.2010.79.

[2] Globus. GridFTP. [Online] 5 2011. http://globus.org/toolkit/docs/3.2/gridftp/, retireved September, 2011

[3] R. Madduri, C. Hood, W. Allcock and W. E, "Reliable file transfer in Grid environments" Proc. Local 27th Annual IEEE Conference on Computer Networks, (LCN 2002), pp. 737-738, doi: 10.1109/LCN.2002.1181855.

[4] V. Velusamy, A. Skjiellum, and A. Kanevski, "Employing an RDMA-based file system for high performance computing" Proc. 12th IEEE International Conference on Networks (ICON 2004), pp. 66-70, doi: 10.1109/ICON.2004.1409089.

[5] Y. Chao-Tung, C. Yao-Chun, Y. Ming-Feng and H. Ching-Hsien, "An Anticipative Recursively-Adjusting Mechanism for Redundant Parallel file transfer in data grids" 13th Asia-Pacific. Computer Systems Architecture Conference (ACSAC 2008), Aug 2008, pp. 1-8, doi: 10.1109/APCSAC.2008.4625456

[6] H. Wei, G. Qi, and L. Jiuxing, "High performance virtual machine migration with RDMA over modern interconnects" IEEE International Conference on Cluster Computing (CLUSTR 2007), Sep 2007 pp. 11-20, doi: 10.1109/CLUSTR.2007.4629212

[7] E. Kwon, J. Park, and S. Kang "Reliable data transfer mechanism on dynamic nodes based overlay multicast" The 7th International Conference on Advanced Communication Technology (ICACT 2005), Feb 2005, pp. 1349 – 1352, doi: 10.1109/ICACT.2005.246199

[8] Deb, Kalyanmoy. Mulit-Objectctive Optimization using Evolutionary Algorithms. Chichester : Wiley, 2004.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-174-8