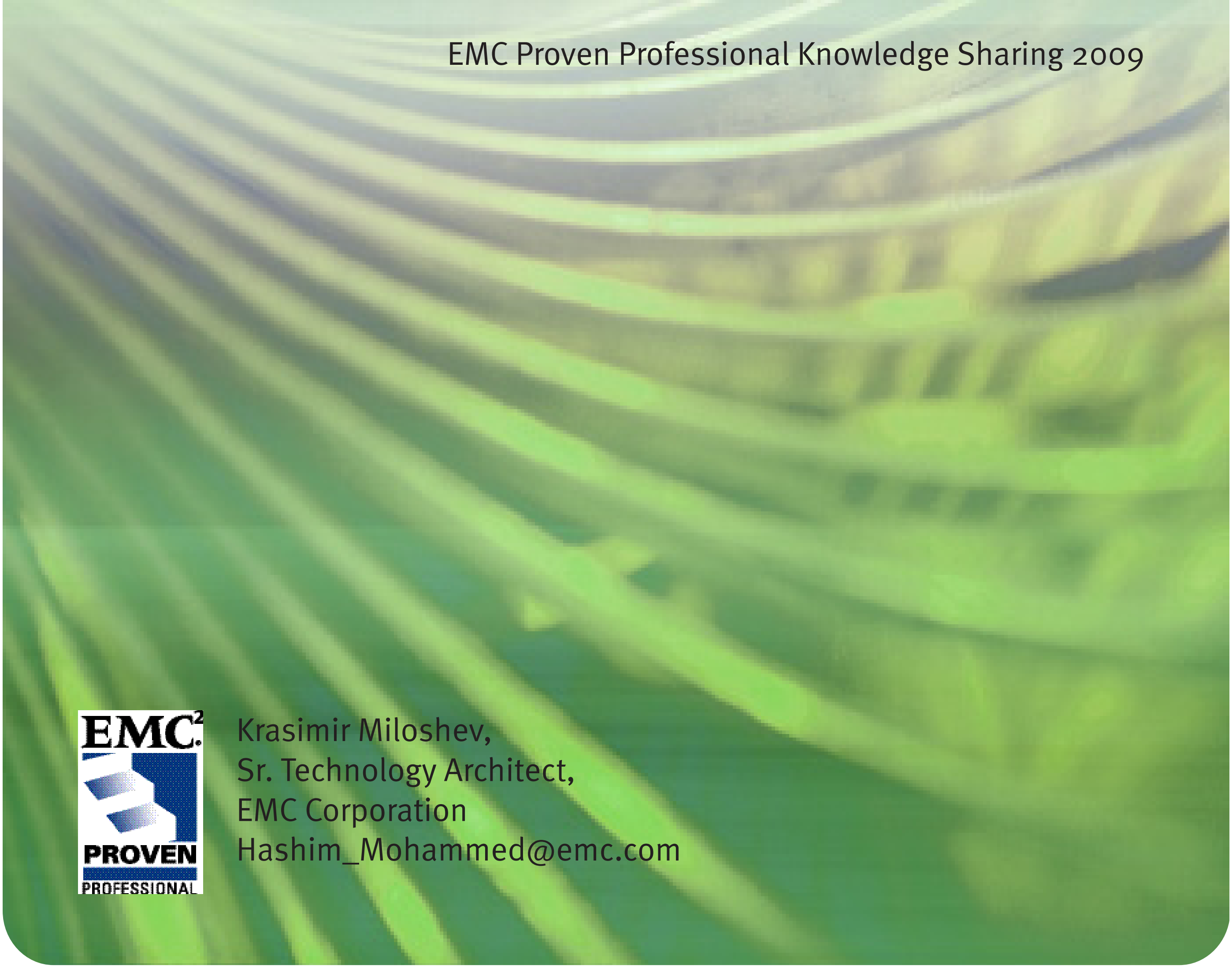
A Load Balancing Algorithm for Assigning Clients to Back-up Media Servers



EMC Proven Professional Knowledge Sharing 2009

Krasimir Miloshev,
Sr. Technology Architect,
EMC Corporation
Hashim_Mohammed@emc.com

Table of Contents

Disclaimer: The views, processes or methodologies published in this compilation are those of the authors. They do not necessarily reflect EMC Corporation's views, processes, or methodologies

Abstract

Finding the optimal backup client's distribution over designated Media Servers can be considered part of the general Load Balancing Problem. Our practical goal is to optimally distribute the backup client's load among designated Media Servers responsible for the backup read/write operations. We can then minimize the Backup Window. The better we balance the client's load, the shorter the backup Window.

Introduction

Here are the basic components of a backup infrastructure (figure 1):

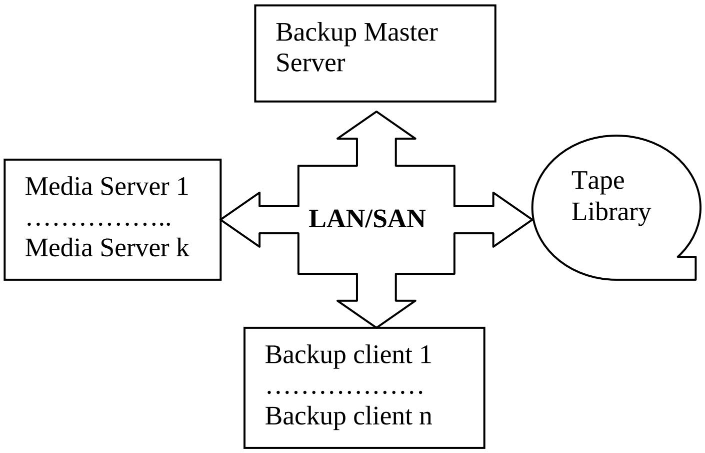
1. **Master server**-the central management and configuration server for the backup/restore operations.

2. **Clients**-these are clients from a backup perspective, but from a functional point of view they are servers – File servers, DB servers, WEB servers etc.

3. **Media servers**- backup the client's data; controlled and managed by the master server. Media servers are often called storage nodes.

4. **Storage backup devices**- usually tape devices/ libraries to store data on.

Figure 1



Assigning proper backup clients to the existing Backup Media Servers is an important task in backup operations. Backup Load, for our purposes, is the amount of data that must be backed up.

The Master Servers can be Media Servers in some small configurations, so we have one combined Master/Media Server. However, in most cases, we have practically separated one Master Server from one or more Media Servers.

Media Servers are a software product residing on a specific host and assisting the Master Server with backup Read/Write operations.

How many Media Servers we should have is not only a question of how big our backup network infrastructure is (how many backup clients we have), but a matter of financial considerations as well. Each Media Server license is costly. There are no recommendations about how many clients should be serviced by one back Media Server so everyone makes decisions based on their own practical considerations.

We would need more Media Servers for larger networks, but the number of clients is not the only factor to determine how many Media Servers we need. From a load perspective, we must consider how much data we have on each backup client, and what type of application runs on the client (file server, DB server etc.).

We will apply a simplified approach based only on the amount of data to be backed up for each of the clients. Our goal is to distribute client loads among the Media Servers in the best possible way. The fewer imbalances between the Media Servers, the shorter Backup Window will get. This means we have to determine how the backup clients are assigned to the Media Servers, and which client “goes” where.

A Load Balancing Schema for Designated Backup Media Servers

We can further simplify the study by investigating a case with only 10 media backup clients and 2 Media Servers. Let us say that the same approach can be applied in cases with larger numbers of clients and media servers. Our goal is to optimally designate the 2 media servers among all the 10 backup clients in order to spread the backup load evenly. Let us present the clients and their loads (in GB) by the next table (fig 2):

Figure 2

clients	1	2	3	4	5	6	7	8	9	10
Load	280	330	100	600	410	750	700	500	80	50

We can sort out these clients by their loads (in GB) and then we will get the next table (figure. 3):

Figure 3

clients	6	7	4	8	5	2	1	3	9	10
Load	750	700	600	500	410	330	280	100	80	50

The next step is to assign the 10 clients to the existing two Media Servers in the best possible way from a load balancing perspective.

The total capacity of data for all 10 backup clients is 3800 GB. Ideally, we would distribute those clients among the two Media Servers so that the amount of data backed up by each Media Servers is 1900 GB.

Let’s introduce a particular load distribution algorithm based on two indexes i and j, and a variable called pivot. Pivot is the sum of all data amounts divided by the number of media servers m. The pivot represents the ideal load to be distributed between the Media Servers.

Here is how the algorithm works; we will use two indexes i and j. On each step "i" increases from bottom up, and "j" decreases from top down. If we have a total capacity of 3800 GB, and since in this example we have two Backup Media Servers, then so called pivot can be estimated by dividing 3800/2=1900.

INDEX_1 and INDEX_2 are two arrays where the distributed client's indexes are placed. Thus we would know which client is assigned to which media server. In our particular case INDEX_1 is related to Media Server 1 and INDEX_2 is related to Media Server 2.

1. S[1] goes to Media Server_1 A and S[2] goes to Media Server_2.

2. For S[10] if S[1] + S[10] < 1900, then S[10] goes to Media Server_1 and S[9] goes to Media Server_2. Otherwise S[10] goes to Media Server_2, not ot to Media Server_1.

3. For S[3] if S[1] + S[10] + S[3] < 1900, then S[3] goes to Media Server_1 and S[4] goes to Media Server_2. Otherwise S[3] goes to Media Server_2, not to Media Server_1.

4. For S[5] if S[1] + S[10] + S[3] + S[5] < 1900, S[5] goes to Media Server_1 and S[6] goes to Media Server_2. Otherwise S[5] goes to Media Server_2, not to Media Server_1.

Here is the final load distribution (figure.4):

Figure 4

Media Server 1	Media Server 2
750	700
50	80
600	500
100	280
410	330

After sorting the backup clients, based on their data load, we grouped those backup clients from the sorted array with the highest and the lowest load and then assigned these consecutive groups of clients to the Media Servers by comparing the total group load with the pivot.

We achieved the following load balancing distribution:

1910 GB load is assigned to Media Server_1 and 1890 GB load is assigned to Media Server_2, which is close to ideal. The ideal case is when we have 1900 GB load evenly distributed over the 2 Media Servers.

Basic Load Balancing Algorithm and Program Implementation

We can easily implement the idea exposed above by creating a load balancing script based on the C or Korn-shell program language.

1. First, sort all the clients by size in decreasing order of their load.

2. Next, distribute these clients between the Media Servers based on the load distribution algorithm presented above.

Here is an implementation of the load distribution algorithm in Korn-shell. Let’s demonstrate the case when we have m=2 Media Servers.

```
# sorting part of the algorithm; array_size is the number of the backup clients;
    for i= 1..array_size-1
      do
          for j=1..array_size-1
           do
           if (S[j] > S[j+1]) then
                   begin
                          temp = S[j+1];
                          S[j+1] = S[j];
                          S[j] = temp;
                   end
            done
       done

# distributing part of the algorithm; m is number of Media Servers
          Media Server_1=0; Media Server_2 =0; m=2
          i=1;j=n;p=1;SUM=0; k=1;
          for p=1..n do SUM=SUM+S[p]
    PIVOT=SUM/m
    While i< j do
       begin
        if Media Server_1 < PIVOT
            then
              begin
                    Media Server_1=: Media Server_1 +S[i] + S[j];
                    INDEX_1 [k]=i;
                    INDEX_1 [k+1]=j;
                    k=k+2;
                    Media Server_2:= Media Server_2+S[i+1]+ S[j-1];
                    INDEX_2 [k]=i+1;
                    INDEX_2 [k+1]=j-1;
                    i=i+2;j=j-2
               end
             else
              begin
                    Media Server_1:= Media Server_1 + S[i]+S[j];
                    INDEX_2 [k]=i;
                    INDEX_2 [k+1]=j;
                    k=k+2
                    Media Server_2:= Media Server_2+S[i+1]+ S[j-1];
                    INDEX_2 [k]=i+1;
                    INDEX_2 [k+1]=j-1; i=i+2;j=j-2
              end
        end
  done
```

INDEX_1 and INDEX_2 are the two arrays where we keep a record of which backup client is assigned to which media Server. INDEX_1 is related to the first Media Server and INDEX_2 is related to the second Media Server. Media Server_1 and Media Server_2 are variable. This is where we keep track of the temporary load of each of the Media Servers. S[i] is an array containing the data loads for each client.

Conclusion

This algorithm distributes the backup client loads among the existing Media Servers. First, we assign the highest load and the lowest load for the fist Media Server, and then we assign the next highest load and the next lowest load to the second Media Server, the next highest load and the next lowest load to the third Media Server etc. If we have 3 media Servers, we will have 3 index arrays INDEX_1, INDEX_2 and INDEX_3 and 3 variables, respectively Media Server_1, Media Server_2 and Media Server_3. You can apply the same approach if you have more than 3 Media Servers.

References

[1] Congias K. The Backup Book, Springer, 2001

[2] Merhra P. and Wah B. Load balancing: An Automated learning Approach, O'Reilly, 2001

[3] Quigley, Ellie UNIX Shells, 1999, Practice Hall PTR, Upper Saddle River, NJ

[4] Wah, Benjamin and Merhra, Pankaj. Load Balancing: An Automated Learning Approach, 2001, O'Reilly Media, Inc.

Biography