AN ADAPTIVE HIERARCHICAL CLUSTERING PROTOCOL FOR MULTIMEDIA OVERLAY MULTICAST APPLICATIONS

Cheng Peng, Dai Qionghai, Wu Qiufeng

Department of Automation
Tsinghua University
100084, Beijing, P.R.C.
chengpeng00@mails.tsinghua.edu.cn

ABSTRACT

Adaptive Hierarchical Clustering Algorithm (AHCA) maps a flat topology to a hierarchical tree though output trees are incontrollable and are not suitable for multimedia. Prune-Relocate operation and Top Topologies operation are proposed in this paper to improve AHCA protocol and generate OM-AHCA trees. Numerical simulations show that OM-AHCA trees are compromise between AHCA trees and single-level topology flat protocol trees, which optimize the overall performance of single-level flat protocol and improve the degree metric of AHCA trees.

1. INTRODUCTION

The Overlay Multicast (OM) protocols use a tree or a group of trees embedded in a mesh to transport the same data to multiple hosts at the same time. Most early OM protocols use single-level mesh, the increasing routing costs and state information maintenance costs make them unsuitable for scalable applications. So Multimedia multicast applications suffer a lot from the lack of scalability and multicast system resource, such as link bandwidth, host storage capability and CPU ability, and so on.

Clustering solutions are used to build hierarchical mesh for better scalability. In [1] a multicast group is described as a hierarchy of clusters by Adaptive Hierarchical Clustering Algorithm (AHCA). In [2] a two-level mesh is constructed by clustering algorithms based on the hosts' Internet distances. DT protocol [3] is a typical single-level overlay multicast topology. Simple "Bounding Boxes" and hierarchical clustering algorithms have been proposed but their performance is poor [3].

However, multimedia applications require multicast protocols to achieve both the scalability and resource utilization. In this paper, based on AHCA, we propose a novel overlay multicast protocol, OM-AHCA, to find a suitable multicast tree. The other parts of the paper is organized as following: Part II reviews the original AHCA algorithm and analyzes its shortcomings; Part III proposes Prune-Relocate operation and Top Topologies operation to make original AHCA trees more suitable for data distribution. Numerical simulation is designed and the result is analyzed in Part IV. Part V gives out the conclusion.

Supported by the Key Project of National Natural Science Foundation of China under Grant No. 60432030

2. REVIEW OF AHCA ALGORTIHM

In AHCA algorithm, any host sees the rest of the world as a set of concentric rings (as shown in Fig. 1) centered on the host itself.

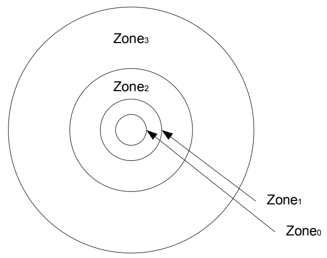


Fig. 1. A set of concentric rings

A parameter, λ, is used to construct the concentric rings (called zones) whose sizes following an "exponential distribution":

$$\begin{array}{l} zone_0 : 0 < dist \leq 1 \\ zone_i : (1+\lambda)^{i-1} < dist \leq (1+\lambda)^i \end{array} \quad (1)$$

with $\lambda > 0$, $i \in N$. The radius of the clusters at $zone_i$ is:

$$r_i = \frac{(1+\lambda)^i - (1+\lambda)^{i-1}}{2} \quad (2)$$

For a certain group, λ influences the clustering results indirectly. Starting AHCA algorithm from a host called the root, the hosts are mapped into a hierarchy of clusters by a

0-7803-9332-5/05/$20.00 ©2005 IEEE

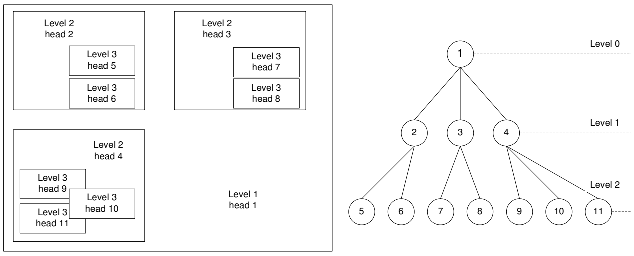


Fig. 2. AHCA clusters and AHCA tree

decision-making process. A cluster is represented by a cluster head and is composed of the cluster head and other hosts "close" to the cluster head. Then the hierarchy of clusters is organized into AHCA tree. All the hosts are included in a cluster whose head is the root, but the bottom clusters which are "singleton-clusters" contain only their cluster heads.

More detailed process can be found in [1]. Set the multicast data source host as the root, use the unicast path's cost as the distance value in AHCA, then run AHCA to achieve the hierarchy of clusters, a source tree which is similar to the one in Fig. 2 can be built to span all hosts. These trees can be called as AHCA trees.

However, the AHCA trees still suffer several weaknesses:

- The clusters built with AHCA are unconstrained, meaning that root cannot explicitly control how many levels the AHCA hierarchy will have. The clusters may be lopsided extremely.
- A host cannot explicitly control the number of members in its own cluster and the number of its children in the tree. This may introduce a terrible degree metric. For those hosts owning many children in the tree very heavy load of replicating and transporting data at the same time is unavoidable.
- The root's problem is especially terrible. If the multicast data is distributed along the AHCA trees, the root host should be required to serve more than 7% hosts of the whole group[1], this may be a visible bottleneck. This may be a significant penalty for trees built for data distribution.
- The turn of a host joining the group influences its position in final output tree. So many different AHCA trees exist for same hosts.

For the above-mentioned reasons, the AHCA trees are more suitable for control purposes. The root has no clear knowledge about the tree's shape. More algorithms should be proposed to address the above issues to make them suitable for data and multimedia distribution though the requirements to multicast trees may change a lot.

3. IMPROVING AHCA FOR MULTIMEDIA DISTRIBUTION

In this section, two algorithms are proposed to make up the original AHCA trees' shortcomings.

Firstly, define all hosts included in the cluster whose head is h_i, except the head itself, as h_i's offspring. The root's offspring is all the other hosts in the group. Define all clusters whose level is 1 as the top clusters, their heads are called top heads.

3.1. Prune-Relocate (PR) operation

Define a parameter Δ. For a host h_j, if the following conditions are met:

1) h_j is in level 2 or lower levels, i.e. it is not the source nor a top head;

2) The number of h_j's offspring is equal to or larger than Δ;

3) The number of h_j's each children's offspring is less than Δ;

h_j's cluster will be called as *super cluster*. The cluster should be pruned from AHCA tree and be relocated as a top cluster. Then h_j becomes the new child of the root and the hierarchy within h_j's cluster is unaltered though they are in higher levels than in original AHCA tree.

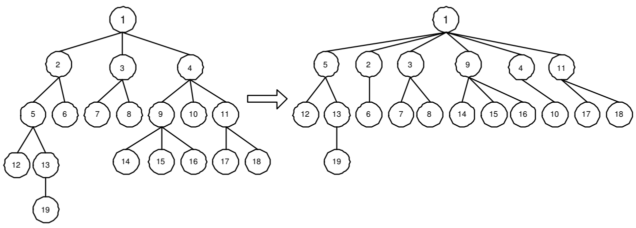


Fig. 3. Prune-Relocate operation when $\Delta = 2$

The Prune-Relocate operation should be started from the lowest level and repeated again and again until no super cluster exists.

As shown in Fig. 3, Δ is set as 2. The original AHCA tree includes 19 hosts. There are 3 super clusters whose head are respectively host 5, host 9 and host 11. These clusters are pruned from the original AHCA tree and are relocated as top clusters. It is obvious that the new tree has no super cluster. We call the AHCA trees after PR operation as PR-AHCA trees.

Prune-Relocate is not equal to classic degree balancing algorithm. In PR-AHCA tree, the degree, the number of a host's children, has not a district and direct bound, but each top cluster has much less members than in AHCA tree. In Fig. 3, there are 6 members in a top cluster averagely before PR operation and only 3 in the tree after PR operation.

The Prune-Relocate rules are simple but useful to reduce the total levels of the tree and to reduce the average number of hosts in one top cluster. It is obvious that the total levels in PR-AHCA tree is not larger than $(\Delta+2)$. For multimedia applications, it is important to reduce the host hops experienced by data. PR operation compressed the AHCA trees while making those roots more clear about the trees' shape.

3.2. Top Topologies (TT) operation

AHCP trees are very compressed. When λ is 1, about 36% hosts may act as top heads in a 500 hosts group.Prune-Relocate operation makes the trees even more compressed. The AHCA tree before PR has 3 top heads in Fig. 3 and the one after PR has 6 top heads. For a group, the source can not take all the top heads as its children in the tree and serve all of them at the same time. So the Star topology of top heads and the root must be adjusted.

Indeed, a random connected topology can reduce the root's degree. However in this paper, similar to ordinary OM protocols, two topologies are built: a Top Mesh and a Top Tree.

Delaunay Triangulations topology and Compass Routing rules are chosen to build the Top Topologies. In Delaunay Triangulations, the average number of a host's neighbors is less than 6 and a well known single-level OM protocol based on Delaunay Triangulations, DT protocol, has been proposed [3]. In DT protocol Compass Routing (CR) [3] is used to build a source tree embedded in the mesh.

The TT operation is similar to DT protocol though the topology does not include all hosts in the group but the top heads and the source. The root and the top heads build Delaunay Triangulations as the Top Mesh. The distance among the source and the top heads may be either geographical distance or *round trip time* (RTT) measured. Then a source tree rooted on the source is built according to CR rules and spans all hosts in Top Mesh.

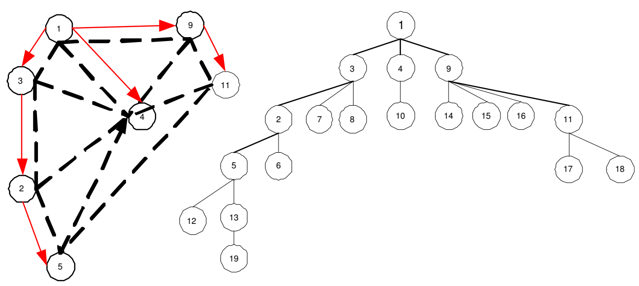


Fig. 4. Top Topologies and OM-AHCA tree

The data is transported along the Top Tree from source to all top heads. When a top head receives multicast data, it replicates and transports them to its offspring along existing branches.

With PR and TT operations, the data distribution topology, OM-AHCA tree, is achieved. The Top Tree may be treated as *trunk* of the OM-AHCA tree. Fig. 4 described the Top Topologies and the final OM-AHCA tree. The thicker branches belong to the Top Tree.

4. SIMULATION AND COMPARISON

The simulation network topology generated by GT-Itm [4] is distributed on a 100×100 plant. There are 8 transit routers, 32 stub routers. One host is connected to a stub router via a link whose cost is 0.3 and delay is 1. The delay of an edge between a pair of routers is equal to its length given out by GT-Itm and its cost is a random value between 0.3 and 1.0. The distance between two hosts used in clustering process is the delay of the unicast path connecting them.

The number of hosts in the group increases from 20 to 100. For a certain group, five trees are built and measured, AHCA tree (λ is set as 1), OM-AHCA without PR, OM-AHCA with PR (Δ is set as 2), DT protocol tree and a tree by random mesh (the overlay link exists between two hosts by a probability of 0.4) and compass routing (RMCR tree).

The metrics to compare the trees are:

- The *Cost* of the tree, which is the total sum of all branches' cost.
- The *Delay* from source to receivers.
- The number of *Total levels* of the multicast tree.
- *Maximum degree* in multicast tree, the extreme situation of degree performance.

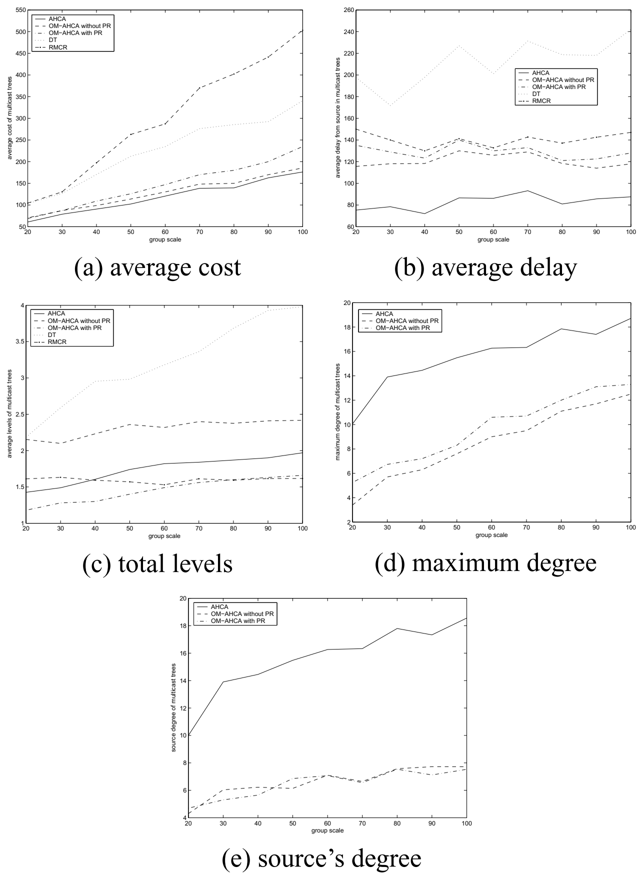
The simulation results are described in Fig. 5.

The results show that all the five kind of trees' cost and delay all increase with the accretion of group. OM-AHCA trees' cost and delay are larger than AHCA trees, but they are all much less than the trees built by DT protocol and RMCR

AHCA trees and OM-AHCA trees have much less levels than those by DT protocol. Though PR operation may introduce more levels, the number of levels of OM-AHCA trees with PR is still less than that of DT protocols.

Some new top heads are produced by PR so the levels in OM-AHCA trees with PR will be less than those without PR, but the cost of the former ones are larger than the later ones. PR operation also increases the average delay from source in OM-AHCA trees.

The maximum degree in OM-AHCA trees with PR is larger than in those without PR. In OM-AHCA trees the maximum number of hosts served by one host is much less than in AHCA trees. The maximum degree metric may



(a) average cost (b) average delay

(c) total levels (d) maximum degree

(e) source's degree

Fig. 5. Simulation results

be reduced more, if PR operation is not executed in OM-AHCA trees.

One special simulation is designed to describe PR's function. Different clustering outputs are achieved for same hosts and same source if they are clustered by different sequence. In above 100-host group, set host 1 as the source, 500 different trees are generated by original AHCA algorithm ($\lambda = 1$) and then PR operation is executed respectively ($\Delta = 2$). Table 1 gives out the comparison of these AHCA trees and PR-AHCA trees. PR operation reduces the variation of trees' cost for same hosts and makes the trees more controllable.

Table 1. The Effects of PR Operation on trees' cost

Algorithm	Average	Variation	90% Confident Interval
AHCA	168.31	88.29	$[154.24, 189.04]$
PR-AHCA	169.14	49.59	$[162.00, 176.84]$

5. CONCLUSION

AHCA is a useful clustering algorithm to map a flat topology to a hierarchical one. It can be treated as a source spanning tree of the multicast group. The AHCA trees use the network resource more efficiently than flat topology protocol, such as DT. However, AHCA almost always introduces a bad degree metric. A host can not control the number of levels in AHCA tree and the number of members in its own cluster in AHCA trees. To make AHCA more suitable to multimedia data distribution, Prune-Relocate operation and Top Topologies operation are proposed.

The numerical simulation shows Prune-Relocate operation is useful to reduce the total levels of the tree and to reduce the average number of hosts in one top cluster. The Top Topologies operation reduces the source's burden of serving too many hosts at the same time which may be very important in multimedia multicast applications. Though PR and TT operation increase AHCA trees' cost, delay and total levels, OM-AHCA trees improve the degree performance of AHCA trees remarkably. OM-AHCA trees have more optimized cost, delay, total levels and maximum degree performance as compared with DT trees.

OM-AHCA trees are compromise between AHCA trees and DT protocol trees. OM-AHCA combine the excellences of AHCA and DT protocols and are suitable for data and multimedia distribution.

6. REFERENCES

[1] L. Mathy, R. Canonico, S. Simpson, D. Hutchison, "Scalable Adaptive Hierarchical Clustering", *IEEE communications letters*, vol. 6, pp. 117-119, March 2002.

[2] Jin JW, K. Nahrstedt, "Large-Scale Service Overlay Networking with Distance-Based Clustering", *PROCEEDINGS LECTURE NOTES IN COMPUTER SCIENCE 2672*, pp. 394-413, 2003.

[3] J. Liebeherr, M. Nahas, Si Weisheng, "Application-layer multicasting with Delaunay triangulation overlays", *IEEE Journal on Selected Areas in Communications*, vol. 20, Iss. 8, pp. 1472 - 1488, October 2002.

[4] E. W. Zegura, K. Calvert, S. Bhattacharjee, "How to Model an Internetwork", *proceedings of IEEE Infocom'96*, San Francisco, pp. 594-602, March 1996.