ISSN:0975-9646

K.Ashok Chakravarthi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (6), 2012,5331-5333

Manageable and Economical Connection of Centralized Repository Servers Using Server Ports

K.Ashok Chakravarthi[1] Khalim Amjad Meerja[2] S.V.Achuta Rao[3]

[1,2]*Department of Computer Science Engineering,*
K.L.University,India

[3]*Department of Computer Science Engineering,*
Vikas College of Engineering,India

ABSTRACT-Objective of centralized repository is to arrange an outsized number of server machines with little apparatus cost while providing high planning facility and bisection width. It is well understand that the present practice where servers are linked by a tree hierarchy of network switches cannot gather these necessities. In this paper, we find out a newest server-interconnection structure. The Structure observe that the commodity server machines used in today's centralized repository generally come with three ports, one for network connection and the remaining for backup purposes. The model supposes that if three ports are aggressively used in network connections, we can construct a scalable, cost-effective interconnection structure without either the luxurious higher-level large switches or any additional hardware on servers. We design such a arrangement structure called Bi Swapped Re Wired Structure Network (BRSN). Although the server node degree is only 3 in this structure, we have proven that BRSN highly scalable to encompass hundreds of thousands of server switch low diameter and high bisection width. We have developed low-overhead traffic-aware routing mechanism to get better effective link utilization based on dynamic traffic state. We have also proposed how to incrementally deploy BRSN.

KEYWORDS-Computer networks, Information processing, Internet, Bi Swapped Re Wired Structure Network (BRSN), Traffic aware routing.

I. INTRODUCTION

Centralized Repository arrangement designs both the network structure and associated protocols to interrelate thousands of servers at a Network Operation Center, with stumpy hardware cost, great and balanced network capacity, and robustness to link sever faults. Its operation is vital to offering both numerous online applications like gaming, Web mail, searching, and infrastructure services.

In this paper, we study an easy technological problem called "Can we build a scalable, low-cost network communications for centralized repository", using only the commodity servers with three ports and low-end, multiport commodity switches. The welfares are First, it costless to build a centralized repository network. Second we do not need high-end expensive switches which are widely used today. Standard, off shelf servers with three ports one for operation in network connection, the other two ports for backup. Third the wiring becomes relatively easy since only three server ports are used for association. Fourth, it may issue more academic research into centralized repository. New problems and solutions in centralized repository arrangement, systems, and applications can be found, implemented, and assessed through an easy-to-build test bed at a university or institution. Today, centralized repository infrastructure may only be afforded by scarce currencies companies such as Yahoo, Juniper. The tree-based solution requires expensive, high-end switches at the top level of the tree in order to alleviate the bandwidth traffic jam. The scaling of the Tree solution is limited to the number of ports at a switch, and it also needs more switches. DCell and BCube typically require ports per server like scaling to large server residents. The basic problem is that, we need to plan a new structure that works for servers with node degree of only 3 in order to scale. The Structure proposes BRSN, a scalable solution that works with servers with three ports only and low-cost commodity switches. BRSN defines a recursive network structure in levels. A high-level BRSN is constructed by many low-level BRSNs. When constructing a higher-level BRSN, the lower-level BRSNs use half of their available backup ports for interconnections and form a structure. This way, the number of servers in BRSN, grows double-exponentially with BRSN levels.

It has three levels of switches. Routing over BRSN is also renovated in two aspects. First, our routing solution balances the usage of different levels of links. Second, BRSN uses *traffic-aware routing* to get better effective link utilization based on dynamic traffic state. In the traffic-aware routing, considering the large server population, we use no central server(s) for traffic scheduling and do not exchange traffic state information among even neighboring servers. Instead, the traffic-aware path is computed hop-by-hop by each intermediate server based on the available bandwidth of its two outgoing links. We have also considered how to incrementally deploy BRSN, which is important for building mega centralized repository. By adding shortcut links in incomplete BRSN, we guarantee the high bisection width of incomplete BRSN. In addition, the shortcut links do not break the routing scheme in BRSN.

In summary, we make three main contributions in BRSN. First, BRSN offers a novel network structure that is highly scalable with off-the-shelf servers of node degree 3 and low-end commodity switches while having low diameter and high bisection width. Second, BRSN uses traffic-aware routing that exploits the available link capacities based on traffic dynamics and balances the usage of different links to get better the overall network throughput. Third, BRSN keeps the merits of high bisection and easy wiring during the incremental deployment by adding shortcut links in incomplete BRSN. BRSN does not offer these appealing features with no cost. The wiring cost is higher compared to the current practice of tree. Besides, servers consume more CPU resources in packet forwarding in BRSN.

II. Related Work

A. Interconnection Structure for Data Centers

We now discuss four interconnection structures proposed for data centers, the current practice of the tree-based structure, and two recent proposals of Tree, DCell, and BCube.

Fat-Tree: In current practice, servers are connected by a tree hierarchy of network switches, with commodity switches at the first level and increasingly larger and more expensive switches at the higher levels. It is well known that this kind of tree structure has many limitations. The top-level switches are the bandwidth bottleneck, and high-end high-speed switches have to be used. Moreover, a high-level switch shows as a single-point failure spot for its sub tree branch. Using redundant switches does not fundamentally solve the problem, but incurs even higher cost.

There are pods each containing two levels of switches, i.e., the edge level and the aggregation level. Each -port switch at the edge level uses ports to connect the servers while using the remaining ports to connect the aggregation level switches in the pod. At the core level, there are port switches, and each switch has one port connecting to one pod.

BRSN differs from Fat-Tree in several aspects. First, BRSN puts the interconnection intelligence on servers, rather than on switches as in Fat-Tree. Second, there are three levels of switches in Fat-Tree, but only one lowest level in BRSN. Hence, the number of used switches is much smaller in BRSN.Consider the total number of servers as and -port switches being used. The number of switches needed in Fat-Tree is , while the number in BRSN is . Therefore, BRSN reduces the cost on switches by 80% compared to Fat-Tree. Third, the number of servers Fat-Tree supports is restricted by the number of switch ports, given the three layers of switches.

DCell: DCell is a new, level-based structure. In DCell , servers are connected to a -port commodity switch. Given servers in a DCell ,The servers in a DCell connect to the other DCell 's, respectively. This way, DCell achieves high scalability and high bisection width. BRSN shares the same design principle as Dcell to place the interconnection intelligence onto servers, but they are different in several aspects. First, the server node degree in a DCell is, but that of BRSN is always 3. As a result, BRSN just needs to use the existing backup port on each server for interconnection and no other hardware cost is introduced on a server. Second, the wiring cost in BRSN is less than that of DCell because each server uses only three ports. Third, routing in BRSN makes a balanced use of links at different levels, which DCell cannot. Finally, traffic-aware routing in BRSN is further designed to exploit the link capacities according to current traffic state.

BCube: BCube is also a novel server-centric interconnection topology, but targets for shipping-container-sized data centers,typically 1 K or 2 K servers. It is also a level-based structure. A BCube is simply servers connecting to an port switch. A BCube is constructed from BCube 's and port switches. More generically, a BCube is constructed from BCube 's and -port switches. Each server in a BCube has ports. Unlike BCube, BRSN is designed for mega data centers. The server node degree of BRSN is constantly 3, while that of BCube is larger. BRSN also has wiring advantage over BCube. In addition, given the same number of servers, the number of switches required for interconnection in BRSN is much less than that in BCube. One downside of BRSN compared to Fat-tree, DCell, and BCube is that BRSN has lower aggregate arrangement capacity. Fat-Tree achieves non block communication between any pair of servers, while DCell and BCube have more ports on a server for routing selection. In fact, the lower arrangement capacity of BRSN results from the lesser number of links and switches, which is the tradeoff for low cost and easy wiring. However, the routing in BRSN we design makes a balanced use of different levels of links and is traffic-aware to better utilize the link capacities.

III. BRSN: A Novel Interconnection Structurefor Data Centers

The Structure present a BRSN physical structure and design the basic routing algorithm on top of BRSN.

A. Networking Rule

BRSN is a recursively distinct construction. A high-level BRSN is constructed by many low-level BRSNs. BRSN is the basic construction unit, which is composed of servers and an n-port commodity switch connecting the servers. Every server in BRSN has one port connected to the switch in BRSN , and we call this port*level-0 port*. The link connecting a level-0 port and the switch is called *level-0 link*. Level-0 port can be regarded as the original operation port on servers in current practice. If the backup port of a server is not connected to another server, we call it an *available backup port*. For instance, there are initially servers each with an available backup port in a BRSN .Now, we focus on how to construct BRSN upon BRSN 's by interconnecting the server backup ports. If there are totally servers with available backup ports in a BRSN , the number of BRSN 's in a BRSN , , is equal to . In each BRSN , servers out of the servers with available backup ports are selected to connect the other BRSN s using their backup ports, each for one BRSN . The selected servers are called *level- servers*, the backup ports of the *level- servers* are called *level- ports* and the links connecting two level- ports are called *leve links*. If we take BRSN as a virtual server, BRSN is infact a mesh over BRSN s connected by level- links. We can use a in order number to recognize a server in BRSN . Assume the total number of servers in a BRSN is N_k,

ALGORITHM

Constructing BRSN upon BRSN 's

```
BRSN  Construction(k){
for(i_1=0 i_1<g_k;i_1++ ; )
for(j_1=i_1*2^k+2^(k-1)-1 ;j_1<N_(k-1);j1==j1 )
   i_2=(j_1-2^(k-1)+1)/2^k+1
   j_2=i_1*2^k+2^(k-1)-1
connect server[i_1,j_1] with[i_2,j_2]
return
}
```

ACKNOWLEDGMENT

We are greatly delighted to place my most profound appreciation to Er.K.Satyanarayana President of K.L.University, Dr.Khalim Meerza Guide, Dr.V.Srikanth Head of the department, under their guidance and encouragement and kindness in giving us the opportunity to carry out the paper. Their pleasure nature, directions, concerns towards us and their readiness to share ideas enthused us and rejuvenated our efforts towards our goal. We also thank the anonymous references of this paper for their valuable comments.

REFERENCES

[1]. S. Ghemawat, H. Gobio, and S. Leungm, “The Google file system,” in Proc. ACM SOSP, 2003, pp. 29–43.

[2]. J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters,” in Proc. OSDI, 2004, pp. 137–150.

[3]. M. Isard, M. Budiu, and Y. Yu et al., “Dryad: Distributed data-parallel programs from sequential building blocks,” in Proc. ACM EuroSys, 2007, pp. 59–72.

[4]. M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in Proc. ACM SIGCOMM, Aug. 2008, pp. 63–74.

[5]. C. Guo, H.Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, “DCell: A scalable and fault-tolerant network structure for data centers,” in Proc. ACM SIGCOMM, Aug. 2008, pp. 75–86.

[6]. C. Guo, G. Lu, D. Li, H.Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, “BCube: A high performance, server-centric network architecture for modular data centers,” in Proc. ACM SIGCOMM, Aug. 2009, pp. 63–74.

[7]. Dan Li,Chuanxiong Guo,”Scalable and Cost-Effective Interconnection of Data-Center Servers Using Dual Server Ports”in IEEE Transactions on Networking,June 2010