http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (8): 1228-1234, 2009
ISSN 1812-5638

An Extended iSCSI Protocol Recognizing Multicast Session: iTRM

Huailiang Tan, Weixin Tang and Bin Yin
School of Computers and Communications, Hunan University, Changsha 410082, China

Abstract: This study presents an extended definition for iSCSI protocol that recognizes multicast session: iTRM (iSCSI transparent reliable multicast) protocol. The iTRM protocol extends the definition of iSCSI PDU in order to interpret multicast session announcement. Sharing data for iSCSI sessions is delivered via multicast session and NAK of multicast session is transmitted by iSCSI session to ensure reliability of multicast transmission. The iTRM protocol adopts a transparent agent that monitors I/O accessing behavior of iSCSI initiators and launches the multicast session when sharing data is requested by several iSCSI initiators. Test results show iTRM protocol improves the performance of parallel I/O operations when initiators boot simultaneously from a single target. iTRM also enhances the stability of I/O performance of iSCSI network computing system.

Key words: Network computing, iSCSI, reliable multicast, iTRM

INTRODUCTION

In traditional von Neumann architecture, computing resources are always bound with storage resources, making computing entities lack the capability of cooperating with each other. To expand the definition of von Neumann architecture, concept of network computing is proposed. Within the network computing system, data resources are stored on-demand deployment equipments, which do not belong to a specific computing entity. Data resources are shared by all network computing entities and computing tasks are handled by several distributed computing entities (Ma *et al*., 2005; Revett *et al*., 2001).

Zhang (2004) built MMNC (Manageable Multi-media Network Computer), which is a network computing system based on Transparence Computing scheme. Test results of MMNC show with the number of host clients increasing, the boot time of OS and applications of hosts grows quickly. So, Zhou *et al*. (2003) introduced multicast protocol as NCBP (network-based client boot protocol) for MMNC system.

MMNC utilizes UDP as transport protocol, which uses stop-and-wait policy and has poor efficiency, making it is unfit for a heavy load network environment. So, Tan and He (2006a, b) build network computing system based on iSCSI protocol (Satran *et al*., 2004), which consists of several host clients (initiators), one storage server (target) and one Gigabit switch. The storage server provides accessing service of data resources such as OS and applications for host clients. The host client has no local storage devices and has to boot from the storage server through its iSCSI HBA. The iSCSI HBA is built by a NIC with ROM consisting of embedded iSCSI/TCP/IP protocol stacks. The host client and storage server communicates via iSCSI I-T session.

Test results of iSCSI network computing system show if there are a few host clients booting from the storage server, the host clients' I/O read throughput can reach or exceed the local PC. However, when many hosts or application programs on hosts start simultaneously, their I/O read throughput decrease.

Because TCP transport protocol is based on end to end link path, when there are several host clients accessing the storage server simultaneously, the server will undertake overburdened load of host clients. As there is high degree of data-sharing of host clients' accession in iSCSI network computing system, it is suitable for sharing data to be transmitted by multicast session in order to conquer weaknesses of end to end link path in iSCSI sessions. For this reason, reliable multicast protocol is introduced into iSCSI network computing system.

RELIABLE MULTICAST PROTOCOL FOR iSCSI SESSION

Within iSCSI network computing system, host clients have no local storage devices, all data is stored on remote storage server and data transmitted to host clients from the storage server has a high-degree of sharing nature. For that case, multicast session of sharing data is feasible for iSCSI network computing system. Lin *et al*. (2007) and Lu *et al*. (2007) applied reliable multicast protocol for data replication in storage area network.

Corresponding Author: Huailiang Tan, College of Computers and Communications, Hunan University, Changsha, 410082, China Tel: 8613975807198

Floyd *et al.* (1997) and Sonoda *et al.* (2001) introduced the design of reliable multicast protocol. Multicast sessions distinguish each other by multicast group address. All multicast receivers must join into a multicast group, which is masked by multicast group address when receiving data. The multicast sender transmits sharing data to multicast router via multicast group address. And the multicast router delivers multicast data to multicast group members. In order to reduce the load of network in contrast with TCP unicast session, multicast sender's network port needs only one link path to several receivers for sharing data. Multicast router is responsible for managing multicast group members and retransmitting data to members of group. To achieve these functions, the IP switch must support IGMP (Internet Group Management Protocol) and MRP (Multicast Routing Protocol) (Wang and Wu, 2003).

Because the transparent protocol of multicast session is based on UDP, which has no guaranty for reliability and order of data package. When host clients receive sharing data, it is impossible to demand host clients to start simultaneously strictly, so that there will be some data lost. As for multicast session, the scheme of retransmission is essential. The scheme of retransmission is based on NAK, which means that the multicast receiver checks the integrity of multicast data and issues a negative acknowledgment to the multicast sender asking for retransmission lost data (Wang and Zhang, 2002; Gemmell *et al.*, 2003).

Within iSCSI multicast session, retransmitting lost data is executed by iSCSI session and the whole process of multicast session and retransmission is transparent to upper SCSI bus, so it doesn't acquire modifying the iSCSI architecture.

ARCHITECTURE OF iTRM

Figure 1 shows the architecture of iTRM. The iTRM needs to be deployed in both client and storage server. It is the iTRM controller between iSCSI bus and SCSI bus and a multicast sender/receiver under the iTRM controller. Within the host client, the iTRM controller integrates unicast data from iSCSI initiator and multicast data from multicast receiver. Within the storage server, the iTRM controller coordinates behavior of iSCSI target and iTRM sender. Within the storage server, there is a multicast monitor agent, which monitors I/O read requests coming from iSCSI initiators. If there are several iSCSI initiators requesting sharing data simultaneously, the multicast monitor agent will command iTRM controller to launch a multicast session.

Considering the heavy load of storage server's network port, it is necessary to deploy two network ports, so that unicast data and multicast data can be split into two individual physical paths. The host client has only one network port and its multicast and unicast data are divided into logical paths.

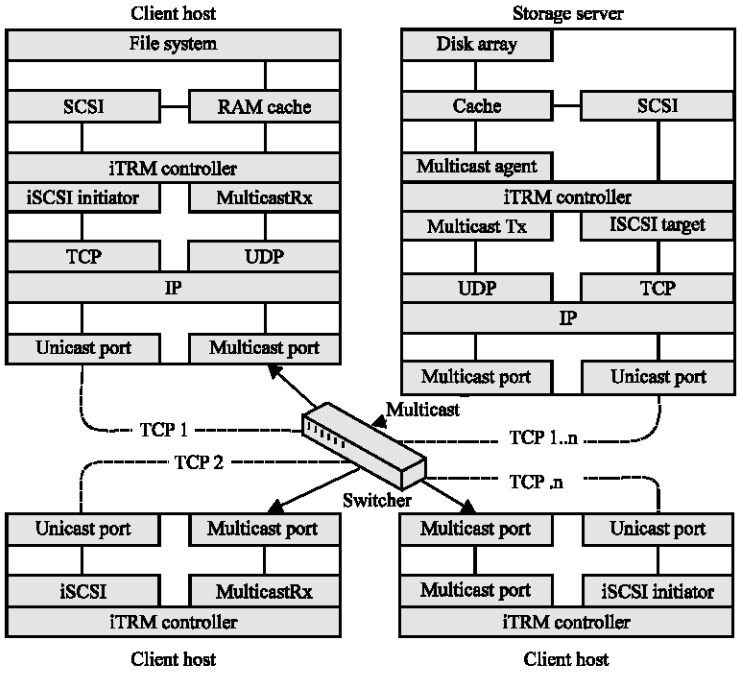

Fig. 1: Architecture of iTRM; sharing data is transmitted by multicast path

When there is no multicast session, the iTRM controller directly transfers data and message between SCSI bus and iSCSI bus, which is the same as normal iSCSI session. When multicast monitor agent decides to launch multicast session, sharing data for iSCSI initiators are transmitted by multicast sender via multicast path and multicast receiver of host client receives sharing data. The iTRM controller of host client is responsible for checking integrity of received data. If there are any lost or incorrect, the iTRM controller commands iSCSI initiator to request lost data from iSCSI target of storage server via iSCSI unicast path.

Within host client, I/O requests from file system are redirected to RAM cache to locate relevant page files. If I/O requests do not hit in RAM cache, the I/O system will call page fault interrupt, then I/O requests will be redirected to iSCSI iTRM control and be transferred to iSCSI bus. iSCSI initiator on host client delivers these I/O requests to iSCSI target on storage server and receives data and responses from storage server. Host client does not care about details of I/O accessing redirections, as it seems like the host client has a local SCSI disk.

Within storage server, iSCSI target receives iSCSI requests from iSCSI initiators and picks up SCSI commands from iSCSI PDU. SCSI commands are transferred to SCSI bus by iTRM controller. iTRM controller tries locating page file in RAM cache for SCSI commands and if it hit in cache, I/O data will be transferred to iSCSI target at first. Otherwise, I/O data will be read to iSCSI target from disk array by SCSI bus and cache will be updated by new data. At the same time, multicast monitor agent is monitoring I/O data in cache. If there are several iSCSI initiators reading sharing data, the multicast monitor agent will inform iTRM controller to launch multicast session.

START POLICY OF MULTICAST MONITOR AGENT

Within storage server, multicast monitor agent starts with the aim of monitoring I/O data in RAM cache and SCSI commands. RAM cache is split into pages by the size of ELBA (Extended Logical Block Address) and one ELBA is composed of 6 SCSI LBAS. Within RAM cache, ELBA is orgamized as linked list indexed by ELBA descriptor header (Tan and He, 2007). Multicast monitor agent generates an accessing table that is composed of triad structures. The triad structure is described as <ELBA descriptor header, initiator name list, access frequency>.

I/O requests and responses are transported between iSCSI initiator and iSCSI target via iSCSI session. When iSCSI request is delivered to storage sever, iSCSI target decapsulates SCSI commands and delivers them to iTRM controller. Then SCSI commands are executed by SCSI bus. The SCSI bus at first reads data from RAM cache. If cache accession does not hit in cache, I/O system will call page fault interruption and SCSI data will be read from disk array and written into cache. So, RAM cache has the latest and most frequently accessed data for iSCSI initiators. Multicast monitor agent indicates the entry address and offset of ELBA for SCSI command by LBA (Logical Block Address) of SCSI CDB (Command Descriptor Block).

DATA PROCESS OF iTRM

Figure 2 shows the process of transmitting data and control messages in iTRM protocol. The full process of iSCSI multicast session is as follows:

- Multicast monitor agent on storage server senses sharing data for iSCSI initiators and commands iTRM controller to launch a multicast session
- iTRM controller notifies iSCSI target that sharing data will be transferred by multicast session. Then iSCSI target sends multicast session announcement to iSCSI initiator, which is a substitute for iSCSI response
- When iSCSI initiator receives multicast session announcement, it cancels previous iSCSI task according to ITT and notifies the iTRM controller of host client to prepare for multicast data receiving
- Host client's iTRM controller activates the multicast receiver, which then joins into multicast group of switch according to multicast address
- iSCSI initiator sends iSCSI multicast session response to iSCSI target to notify that host client get ready for multicast data
- iTRM controller of storage commands the multicast sender to transmit sharing data to multicast group address of switch
- Switch retransmits sharing data to members of multicast group
- iTRM controller of host client integrates multicast data and checks integrity on the basis of SCSI LBA. If there are data lost or incorrect, iTRM controller commands iSCSI initiator to send iSCSI request to iSCSI target for retransmission
- iSCSI target transmits lost data according to SCSI LBA in iSCSI requests and the process is the same as normal iSCSI session
- iTRM controller integrates data from multicast session and iSCSI unicast session and transfers the integrated data to upper file system

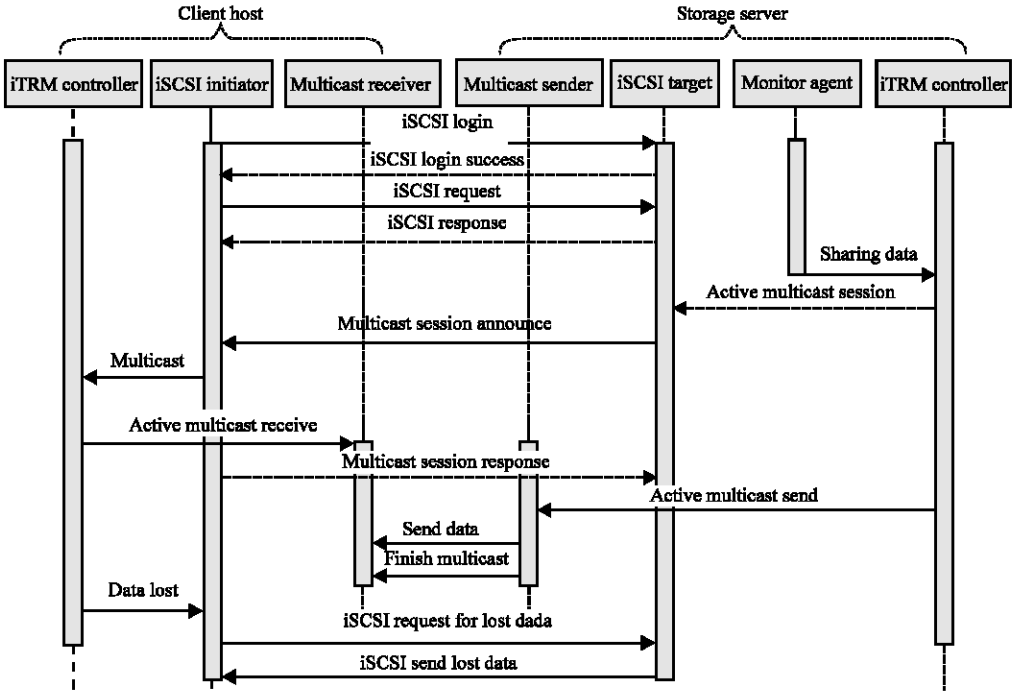

Fig. 2: Data and control messages processing in iTRM

EXTENDED DEFINITION OF iSCSI PDU

The classic definition of iSCSI protocol does not support multicast session, it is necessary to extend the definition of classic iSCSI PDU for the iSCSI session to recognize multicast session announcements and responses and to execute coordinate operations.

Opcode of iSCSI PDU is defined in base header of PDU. Since, iSCSI protocol has provided a vender opcode, it is not necessary to modify the structure of iSCSI PDU and the further work is only to add multicast session announcement and response opcodes. The multicast session announcement transferred from iSCSI target to initiator is defined as 0x3c. The multicast session response, which is the response of multicast session announcement and is transferred from iSCSI initiator to target, is defined as 0x1c.

The structure of iSCSI multicast session announcement PDU is only composed of base header. Its total length is 48 bytes and the length of attach header and data segment is zero. The iSCSI multicast session announcement PDU is composed of ITT (initiator task tag), multicast group address, entry address of ELBA, length of multicast data and MSSN (Multicast session sequence number). Figure 3 shows the specific definition of multicast session announcement PDU. The ITT field contains the initiator task tag, which identifies relevant iSCSI request and tells initiator that relevant iSCSI task will be completed by multicast session. MSSN (Multicast Session Sequence Number) is used to identify multicast session announcement for multicast session response. ISID (Initiator Session Identify) and TSIH (Target Session Identify Handle) is used to identify iSCSI session.

The structure of iSCSI multicast session response PDU is only composed of base header too. The length of both attach header and data segment is zero and the total length of multicast response PDU is 48 bytes. The header of the multicast session response PDU contains multicast response and status fields, which are the response and status for multicast session announcement. Figure 4 shows the specific definition of multicast session response PDU. After receiving multicast session announcement from iSCSI target, the host client has to do a series of work preparing for multicast receiving, such as canceling previous iSCSI task, applying for receiving buffer and activating multicast receiver thread. If preparing work fails, the response field tells iSCSI target that host client rejects executing multicast receiving operation and status field show reasons of rejection.

When iSCSI target has received multicast session responses from all members of multicast group, or the timer of multicast session announcement is expired, iTRM controller demands multicast sender to transmit sharing data to multicast address. Then the sharing data will be retransmitted to members of multicast group by switch.

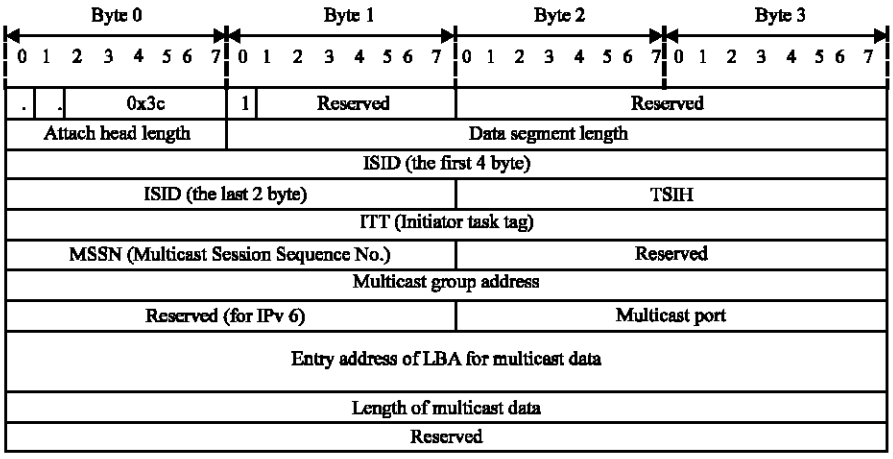

Fig. 3: Definition of iSCSI PDU for multicast session announcement

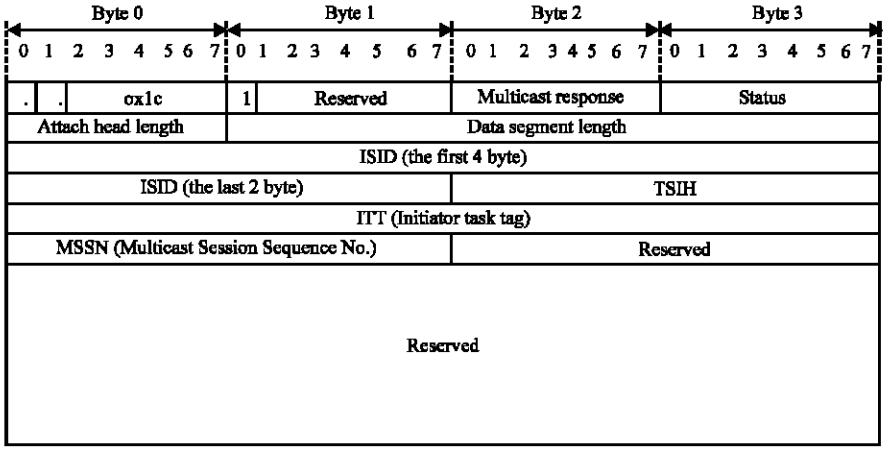

Fig. 4: Definition of iSCSI PDU for multicast session response

If multicast session announcement of iSCSI target is rejected or not recognized by iSCSI initiator, iTRM controller of storage server will cancel multicast session for the iSCSI initiator and continues previous iSCSI session to transfer data to host client.

RESULTS

The iTRM protocol is an I/O acceleration technology for iSCSI network computing system, so the test for iTRM adopts contrast experiments, which include the OS boot time and application start time of host clients with or without iTRM protocol. We established a network computing system consisted of one storage server and 32 host clients. Host clients have no local storage disk and are connected to storage server via 1000 Megabytes Ethernet network card. Having adopted iSCSI remote booting technology, host clients could use network card as iSCSI HBA to remotely boot OS or applications which are stored on storage server.

In our experiments, we examined the fluctuations of remote parallel I/O performance when the number of host clients booting simultaneously grew. We choose Windows XP Professional SP3 to test OS boot time and choose Photoshop 7 to test application start time. Figure 5 and 6 contrastively show the average OS boot time and application start time while with or without iTRM protocol (the first column of each figure denotes the OS/application boot time of PC boots from local disk for contrast). The two figures distinctly indicate that without iTRM protocol, OS boot time and application start time of host clients increases quickly as the number of host clients grows. While, introducing iTRM protocol, OS boot time and application start time of host clients increase relatively slow. If the number of host clients surpasses 32, the effect of I/O acceleration of iTRM is obvious.

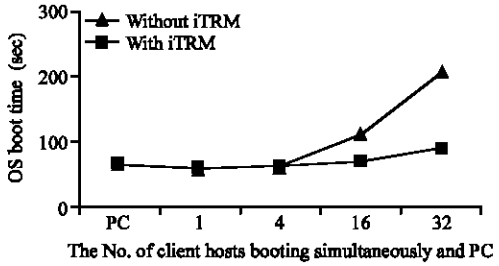

Fig. 5: OS start time with/without iTRM

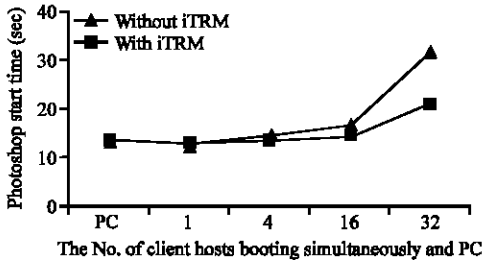

Fig. 6: Application start time with/without iTRM

DISCUSSION

As iSCSI network computing system is based on the method dividing storage and computing, data mustered on storage server and is shared by host clients. Host clients in the same type (use the same OS or applications) share their data (except private data). When these clients boot from storage server simultaneously, the server's network ports actually send many copies of the shared data to different TCP links. After introducing iTRM for I/O acceleration, storage server transport data to host clients using transparent multicast technology and the server's network ports merely send one copy of the shared data. Consequently, the load of storage server's network ports won't quickly increases when the number of host clients grows and the expansibility of iSCSI network computing system is enhanced. Test results show iTRM protocol improves the performance of parallel I/O operations when clients boot simultaneously from a single server.

If host clients do not start the OS or application simultaneously, the effect of I/O acceleration of iTRM is not so obviously. However, iTRM protocol enhances the stability of I/O performance of iSCSI network computing system when there are numerous sudden I/O requests coming from several host clients.

ACKNOWLEDGMENTS

This study was supported by grants from the Research Fund for the Doctoral Program of Higher Education (No. 200805321029) and Municipal Natural Science Foundation of Hunan Province of China (No. 07JJ6139).

REFERENCES

Floyd, S., V. Jacobson, C.G. Liu, S. McCanne and LX. Zhang, 1997. A reliable multicast framework for light-weight sessions and application level framing. IEEE/ACM, 5: 784-803.

Gemmell, J., T. Montgomery, T. Speakman and J. Crowcroft, 2003. The PGM reliable multicast protocol. IEEE, 17: 16-22.

Lin, S.B., M.H. Lu and T.C. Chiueh, 2007. Transparent reliable multicast for ethernet-based storage area networks. Proceedings of the 6th IEEE International Symposium on Network Computing and Applications, Jul. 12-14, Cambridge, MA, USA, IEEE Computer Society, pp: 87-94.

Lu, M.H., S.B. Lin and T.C. Chiueh, 2007. Efficient logging and replication techniques for comprehensive data protection. Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies, Sep. 24-27, California, USA, IEEE Computer Society, pp: 171-184.

Ma, Y.L., X.L. Fu, X.M. Han and L. Xu, 2005. The separation between storage and computation. J. Comput. Res. Dev., 42: 520-530.

Revett, M., I. Boyd and C. Stephens, 2001. Network computing: a tutorial review. Electronics Commu. Eng., 13: 5-15.

Satran, J., K. Meth, C. Sapuntzakis, M. Chadalapaka and E. Zeidner, 2004. Internet small computer systems interface (iSCSI). IETF. RFC 3720. http://portal.acm.org/citation.cfm?id=RFC3720.

Sonoda, M., K.J. Okamura and K. Araki, 2001. Design of general reliable multicast architecture with active network framework. Proceedings of the 15th International Conference on Information Networking, Jan. 31-Feb. 2, New York, USA, IEEE Computer Society, pp: 825-830.

Tan, H.L. and Z.H. He, 2006a. The driver method of virtual SCSI HBA on ISCSI network storage protocol. Comput. Technol. Automation, 25: 57-60.

Tan, H.L. and Z.H. He, 2006b. A TCP implementation on an embedded bare computer platform. J. Hunan Univ., 33: 119-124.

Tan, H.L. and Z.H. He, 2007. Design of storage system cache based on RAID50. Comput. Eng., 33: 215-218.

Wang, X. and S.J. Zhang, 2002. Research on reliable IP multicast transport protocol RMSA. J. Tongji Univ., 30: 475-478.

Wang, J. and Z.M. Wu, 2003. Multicast protocol over switch ethernet. J. Software, 14: 496-502.

Zhang, Y.X., 2004. Transparence computing: concept, architecture and example. Acta Electronica Sinica, 32: 169-174.

Zhou, Y.Z., Y.X. Zhang and Y. Wang, 2003. A customizable boot protocol for network computing. J. Software, 14: 538-546.