ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Implementation of Command Transfer Protocol for Cluster in Networking

Ms.S.Latha*, Ms.A.Nirmala Devi & Ms.S.Gowri

* Department of Computer Science, K.S.Rangasamy College of Arts and Science, Tiruchengode, India

s.latha@ksrcas.edu

Abstract

Computational cluster is used for heavy computation that emphasizes various requirements for network functionality. Most of the computing software uses TCP/IP for their computation. But it has lot of disadvantages. This paper proposes a specialized networking protocol based on cluster computation called Command Transfer Protocol (CTP) which is used for message transfer and heavy computation between computers in cluster. Here group of computers are called clusters and messages sent between the clusters are called commands. CTP uses various functionalities which are absent in UDP and it is more efficient than TCP. This paper implements CTP in message transfer between computers and compares it with TCP/IP and UDP.

Keywords: Computation, Command Transfer Protocol (CTP), Transmission Control Protocol/Internet Protocol (TCP/IP), User Datagram Protocol (UDP).

Introduction

Cluster is a collection of workstation which performs some specific computations. It is a specific system that state special requirements for network functionality. A good quality cluster must have fast data interchange, Reliable data transfer, Broadcasting support; huge data blocks interchange, Peer to Peer networking [1].

About TCP

It is referred as Transmission Control Protocol. It provides reliable data exchange between one system to another. Among its features, TCP controls message size, the rate at which messages are exchanged, and network traffic congestion. As for **IP**, it handles lower-level transmissions from computer to computer as a message transferred across the Internet[4].

About UDP

It is part of the base protocols of the Internet Protocol Suite. Programs on networked computers can send short messages sometimes called datagrams. UDP does not guarantee any reliability (it happens that datagram may be out of order, duplicated, or goes missing without any notice). The fact that no checking whether all packets are actually delivered is made, UDP proves to be faster and more efficient, for applications that do not need guaranteed delivery. UDP is used in the following situations:

- Time-sensitive applications. The problems due to delayed packets are avoided.
- It is also useful for servers that answer small queries from huge numbers of clients. UDP supports packet broadcast (conveys to all on local network) and multicasting (conveys to all subscribers).

Difference between TCP and UDP

TCP is connection oriented protocol whereas UDP is connectionless protocol. When a message is sent along TCP, it will reach the destination unless there is failure. If there is any failure, the sever will request the lost part. So reliability is maintained. But in UDP when a message is sent, it will not be known until it reaches the destination and there is a possibility of message loss. The order of the data is easily known in TCP but it is difficult to predict in UDP. TCP is a Heavyweight protocol whereas UDP is Lightweight protocol. In TCP the packets are sent in streaming order but in UDP packets are sent individually [3].

S.No	TCP	UDP
1.	Connection Oriented Protocol	Connectionless Protocol
2.	Reliability is maintained	Possibility of data loss
3.	Order of the data is easily known	Difficult to predict the order of data
4.	Heavyweight protocol	Light weight Protocol
5.	Packets are sent in streaming order	Packets are sent individually

Table 1 Difference between TCP and UDP

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Disadvantages of TCP/IP

Parallel computing software uses TCP/IP but it has lot of disadvantages. Low speed of data interchange. The "reliability" and "universality" of TCP has a lot of overhead charges. TCP does not support broadcasting multicast connections. Ideology of logical channel creation before data interchange is redundant for cluster computations. Firstly because cluster, as usual, is a well tuned, good working net. Secondly because, some strategies of cluster computing lead to disordered interchange between workstations. TCP is a stream-based protocol, but, for determined tasks, bounded blocks interchange is preferable, because it allows to say definitely, when all data, necessary for further operations, have arrived.

Disadvantages of UDP

UDP is not reliable and the size of UDP datagram is limited by 65467 bytes. There are no guarantees with UDP. It is possible that a packet may not be delivered, or delivered twice, or delivered not in time [2]. Data has to be divided manually.

Need of CTP

To overcome the disadvantages of TCP and UDP, a specialized protocol is used for cluster computing. So CTP protocol is used for faster message transfer in heavy computation.

Implementation

A packet which has header and empty body is called confirmation, which has three differences from headers of the packet having been confirmed. In confirmation header the following condition to be applied Packet's size is set to header's size, in command's number, highest bit is set and Message size is set to zero. It is not a proper solution to confirm each packet with a different confirmation. First thing the recipient does when it receives a packet is it checks if the same packet has been already received. If such a packet was already received, that means that the sender failed to get the confirmation, so confirmation has to be sent again and this packet's receiving procedure can be skipped.

Confirmation has to be, exactly, "sent again", not "resent". Confirmations are not stored in the sent packets storage, but are generated when needed. If such a packet has not been received earlier, then information about its arrival needs to be stored. If the got packet represent the normal command, then the server informs the application about data arrival. If it is a part of a large command, then the server stores it to the large command storage. If it is the last remainder part of the message, then delivery also is to be generated. After the packet has found its place, confirmation has to be sent, and the recipient begins to wait for the next packet. When the sender receives any confirmation, it is to delete the corresponding record from the sent packets storage. The mechanism, like a physical system, aspires to minimize its potential energy, to free all storages as soon as possible.

The CTP implementation contains four storages for data which are accumulated during the lifetime and it provides functionality.

1. It stores session information such as next packet id, description of each packet and interchange time out. If the packet is not confirmed then interchange timeout is used to identify when the packet should be resend. An error message will be displayed if the packet is not confirmed after 8 resending. The default value of timeout is 100 ms. If the timeout value is zero then it will get switched off.
2. Before sending command the packets have to be arranged and memory to be allocated for each packet. There is a storage area called sent command storage where the packets are stored and it will get removed after receiving confirmation.
3. After receiving each and every part of the message, it is stored in buffer and later it is arranged and intimation will be given to server regarding the data retrieval.
4. There is a list called delivery list where all the received messages are stored.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

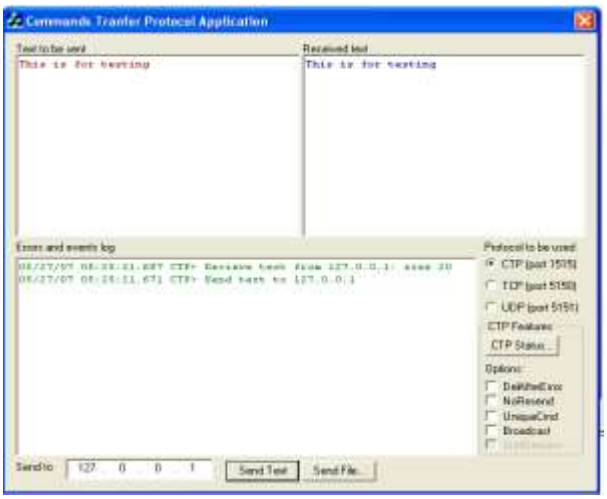

Figure 1 Message passing using CTP

The above diagram shows the message passing using Command Transfer Protocol from source site to destination site.

The below diagram shows status information like sessioninfo, sent commands, large commands, deliveries and various threads when a message is passed using CTP.

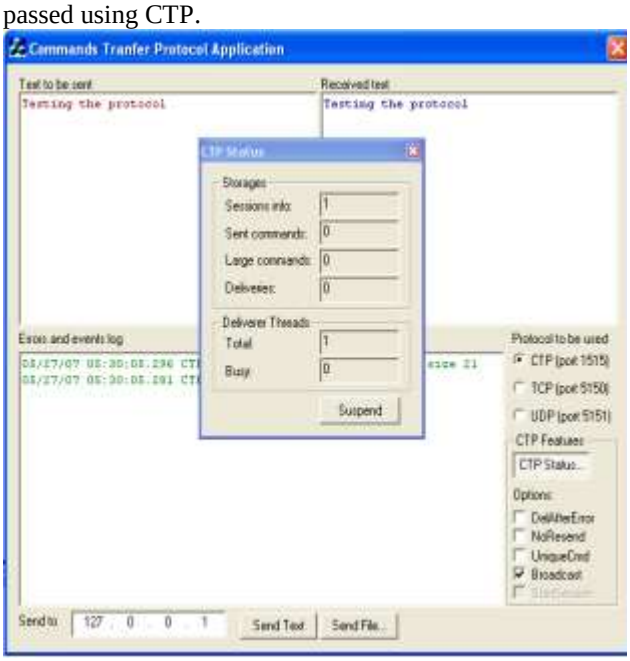

Figure 2 Status message in CTP

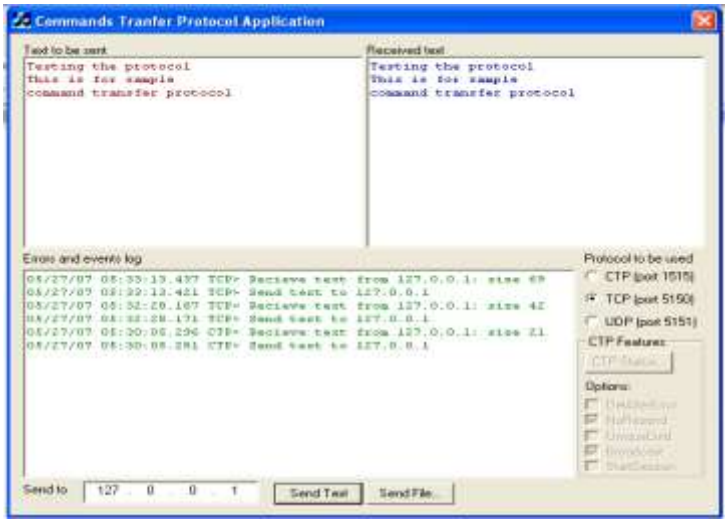

Figure 3 Message passing using TCP

The above diagram shows the message passing using TCP.

Future work

The problems occurred during the protocol support can be saved using UDP. In sender side the identification numbers can be used with the combination of integer numbers and alphabets. Each sender must have unique ID. The header format can be enhanced. In future it can handle large volume of data blocks, improve packet storage and timing performance.

Conclusion

Command transfer protocol assures the networking requirements to increase the interchange speed. Reliable data interchange is applied. The packet which is sent will be stored until it receives the acknowledgement from the recipient. To maintain this method, packets are to be stored with identifiers by assigning integer numbers on the sender side. These identifiers must be unique for each sender. It supports high volume of data interchange. If a message is greater than 65400 bytes, it is separated into smaller parts and these parts will be sent to the receiver one by one. On the recipient's side the parts will be arranged into single message. In receiver side, if all the parts of the message have been received then it gets the information about command's arrival, such commands are known as large commands. Client and server functionalities are implemented in CTP. While working with normal commands CTP is too faster than TCP. CTP is used as a networking mechanism for clusters.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

References

1. http://www.codeproject.com/Articles/6742/Commands-Transfer-Protocol-CTP-A-New-Networking-Pr
2. http://smblog.iiitd.com/2010/09/advantages-and-disadvantages-of-tcp-and.html
3. http://www.diffen.com/difference/TCP_vs_UDP
4. http://en.kioskea.net/faq/608-differences-between-the-udp-and-tcp-protocols