Converse and Charm++ Libraries Version 1.0

June 22, 2000

Contents

1 Introduction

This manual describes CHARM++and Converse libraries.[1]

For a description of the C-based CHARM parallel programming system, please refer to the *Charm Programming Language Manual* and the *Tutorial Introduction to Charm*[1].

2 Irecv Library

IRECV library provides asynchronous communication mode for chare array. The program breaks into two parts("split phase" structure): nonblocking send, receives and iwait, and callback functions. It provides a style that mpi programmers may find intuitive.

There are three functions in Irecv library.

1. void send(void *buf, int size, int dest, int tag, int refno); send function send message which is pointed by "buf" to another array element whose index is specified by "dest" with tag and refno. "buf" is a message buffer containing the data to be sent; "size" is the total size of the message in byte. Like in MPI_send, the tag is used for matching message with dest. The integer "refno" is a reference number, usually the iteration number.

2. void irecv(void *buf, int size, int source, int tag, int refno); irecv function registers tag and pointer to the library. When the desired message arrives, it copies the matching message(with tags) into the location given by the "pointer".

3. void iwaitAll(recvCallBack f, void *data, int refno); Wait function waits for all the previously issued irecvs to complete, and then it calls the callback function "f" with "data" as argument.

To use Irecv library, first create a chare array which is inherited from class "receiver". The sender prepare the message buffer and call "send" function to send message to another array element; The receiver specifes the matching tags and buffer to get the message. After irecv, receive side need to call iwaitAll function to wait all the receive function calls complete. However, the iwaitAll is a nonblocking function. The callback functions will be called after receive calls complete.

Here is an example:

```
int size = 100;
for (int i=0; i<size; i++) buf[i] = data[i];
send(buf, size*sizeof(double), neighbor, tag, iter);

irecv(buf, size*sizeof(double), neighbor, tag, iter);

iwaitAll(callfunc, this, iter);
```

and callback function can be declared as:

```
void callfunc(void *obj)
{
}
```

[1]CHARM is no longer actively supported and maintained, and these manuals are kept only for offering the historical perspectives.

References

[1] L. V. Kale, Milind Bhandarkar, Narain Jagathesan, Sanjeev Krishnan, and Joshua Yelon. Converse: An Interoperable Framework for Parallel Programming. In *Proceedings of the 10th International Parallel Processing Symposium*, pages 212–217, April 1996.