Software Architecture

- principle and practice

第四讲　分布式体系结构基础技术

        - receive(addr,&mptr)告诉它所在的机器上的内核，用地址addr 侦听，并准备去接收发送给addr的一个消息，当有消息到达后，如果非该地址的消息则抛弃，否则接收进程的内核将它拷贝到mptr所指示的缓冲区，并唤醒该接收进程。
    - 特点
        - 一个地址联系一个进程；
        - 若服务器调用在客户调用send之前，非常有效。
        - 如果send原语在receive之前已经完成，则服务器内核就不知道到达的消息拷贝何处。只好抛弃这个非期望的消息，并采用超时重传机制。
        - 假定有两个或更多个客户使用同一个服务器，当服务器从他们之中的一个客户收到一个消息后，在它未完成处理之工作之前，不再用这个地址去侦听发送，在它返回到循环顶部重新调用receive之前，由于要进行相应的处理工作需要花费一些时间而其他客户可能多次尝试发送消息给它，并且有些客户可能放弃。在这种情况下，客户是否继续重发，取决于重 发计时器的值以及他们的忍耐程度。
        - 消息丢失的机会大。
- 非缓冲通信原语 - 消息保留一段时间(keep incoming messages for a little while)
    - 含义
        - 让接收内核对传送来的消息保留一段时间，以便使得一个合适的receice原语能很快地接收：一个非期望的消息到来时，就启动一个计时器，如果计时器在走完之前仍无合适的receive出现，就将该消息丢弃。
    - 特点
        - 减少消息丢失的机会；
        - 如何存储和管理过早到达的消息存在问题。
- 缓冲通信原语 - 信箱(mailbox)
    - 一个希望接收消息的进程要告诉内核为其创建一个信箱，并在网络消息包中指定一个可供查询的地址，然后，包含这个地址的所有到达的消息都被放进信箱。现在调用receive就可以从信箱中取出一个消息，如果没有消息就阻塞（假定用阻塞原语)。
    - 特点
        - 解决了消息丢失和客户放弃消息所引起的一系列问题；
        - 信箱是有限的，有可能装满而造成消息的丢失。
- 不可靠(Unreliable)原语
    - 不可靠send原语指：系统不保证消息能被对方正确接收，要想实现可靠的的通信，由用户自己完成。
- 可靠(Reliable)原语
    - REQ-ACK-REP-ACK
        - 客户向服务器发送一个请求(REQ)，服务器对这一请求由内核向客户机内核返回一个确认（ACK），当客户机内核接收到这一确认后，释放这一用户进程。
        - 说明
            - 确认(ACK）只在内核间传递，客户和服务器进程都看不到这样的过程；
            - 服务器到客户的应答也存在REP-ACK的过程。
    - REQ-REP-ACK
    - 客户在发送一个消息后被阻塞，服务器内核并不返回一个确认(ACK)，而是利用应答(REP)本身作为一个确认(ACK)，客户如果等待时间太长，其内核可重发一个请求，以防请求消息丢失。客户在收到应答(REP)后，向服务器发送一个确认(ACK)，服务器在发送应答后处于阻塞状态，直到确认(ACK from client's kernel)，服务器也可以采用超时重传机制。
    - REQ-(ACK-)REP-ACK
        - 是一种折衷的方案： 当一个请求到达服务器内核时，就启动一个定时器，如果服务器送回应答足够快(计时器时间未到)则应答(REP)充当一个确认(ACK)；如果定时器时间到就送一个单独的确认(ACK)。于是大多数情况下，仅需要两条消息：请求和应答，在复杂情况下，需要另一个确认(Acknowledgement)消息。