**Linkage**

* Purpose And Background

When establish a big Web Site, it's inevitable to use the distribution technology. After we divide the whole system into distributed part, we could call them as services each other. Each service could provide the specific function like Sending Message to the user when Login, Writting the record into the database.

But how to call a distributed service ? Is there a need that we establish a protocol for each service calling. Answer is definitely not, we need to design a unified middleware used between the service and the service caller.

Linkage is such a middleware which could bond all of the distributed services into one whole system. See the chart below, applications, services, db, cache are placed at distributed computers. Between each is linkage. We use linkage to deal with the communication with services and applications.

Application1

Application2

Application3

Service 1

Service 2

Service 3

DB

File System

Cache

* Overall Design

Chart below show the main structure of the linkage. Client side & Service side will communicate with each other by different type of io. NIO is now most widely used, so we use this type of io. But still keep the extension for other type of io. The wapper layer placed above the io layer. In this layer, we will wrapper/unwrapper message with the io protocol. The serialization/deserialization layer placed above the wrapper layer, it will use the serialization protocol. Above this layer is the service access/provider layer. In this layer, we could also define the calling and being called protocol.

Communication

Message Wrapper

Message Unwrapper

Communication

Serialization/Deserialization

ServiceAccess

Service Provider

Serialization/Deserialization

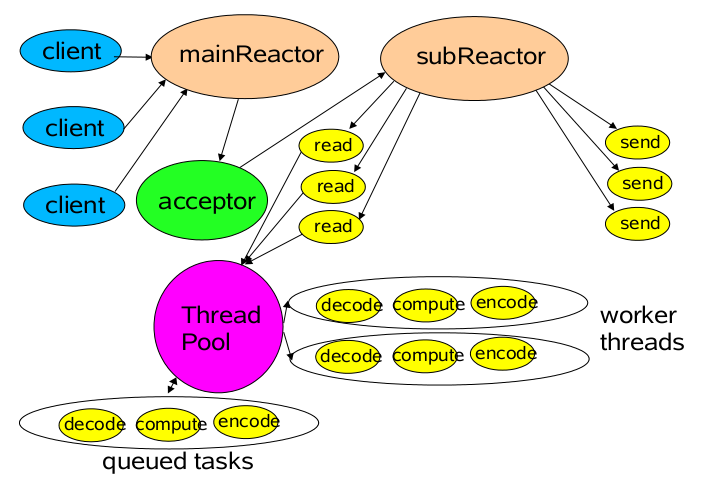
Client Side

Service Side

Linkage

* Communication Layer

Below chart shows the classic NIO model. There is one main reactor accept all connection then the acceptor will post the connect to the sub reactor. Sub reactor will deal with the read&write operation of the connection. Netty actually use this model for its nio case. We would use this model too to deal with more concurrent requrest.



* [Performance](javascript:void(0);) [index](javascript:void(0);)

Concurrence and presure test