优化方案说明

一、Kitex Client Provider 缓存策略

缓存提高性能

```go
//缓存更新策略
func (ptr *defaultKitexClientProvider) dynamicClientCacheRefresh() {
        for {
                time.Sleep(ptr.updateInterval)
                ptr.cache.lock.Lock()
                fmt.Println("dynamicClientCacheRefresh lock")
                var ch chan chObj = make(chan chObj, ptr.cache.ls.Len())
                var taskCount int
                for k := range ptr.cache.data {
                        go ptr.fetchIDLContent(k, ch)
                        taskCount++
                }
                for i := 0; i < taskCount; i++ {
                        chObj := <-ch
                        if chObj.idlContent != "" {
                                ptr.cache.data[chObj.serviceName].Value.(*cacheD
                                fmt.Println("update idlContent for serviceName:"
                        } else {
                                //idl管理平台无数据返回，说明服务已下线或出错，则删除
                                ptr.cache.delete(chObj.serviceName)
                        }
                }
                fmt.Println("dynamicClientCacheRefresh unlock")
                ptr.cache.lock.Unlock()
        }
}

func (ptr *defaultKitexClientProvider) fetchIDLContent(serviceName string, ch ch
        idlContent := ptr.idlProvider.FindIDLByServiceName(serviceName)
        ch <- chObj{serviceName: serviceName, idlContent: idlContent}
}

.......
```

二、IDL Provider 缓存策略

缓存提高性能

```

```