

# 作业2：分布式组成员服务报告

---

## 1.小组成员

---

2052710 吴笛

2052717 陈晨

## 2.项目结构

---

### 包结构

项目由java构建，分为client/common/entity/server四个包

#### client

包括QueryClient类。此类是查询客户端的主类，客户端通过运行这个类，执行日志的搜索查询。

#### common

包括ServerInfo类，ServerInfo类负责存储一些项目中规定的常量。

#### entity

包括Logger与MemberList类。Logger类负责日志的生成，添加，查询。MemberList类存的是组成员结点的信息。

#### server

包括CommonServer、Daemon、Introducer、Query四个类。CommonServer是普通的组成员服务器的主类，Daemon是每个服务器运行的后台程序，负责探测并维护组成员列表，introducer是负责处理新成员加入的结点，Query是服务端执行查询日志的类。

## 逻辑架构

本题的逻辑架构如下：

对于普通的服务器结点，启动主类（commonServer）后会开启并运行query与daemon线程，其中daemon线程会运行sendBeating、receiveBeating、gossiping等线程，完成通过heart-heating、gossiping协议来维护与更新组成员列表。而query线程则会等待client端发来请求，进行日志的查询与返回。

对于introducer结点，不仅要启动commonServer进程，还需要启动introducer进程。introducer进程专门负责结点的加入工作。若introducer故障，则新结点无法加入组，但这不会影响daemon线程的功能。

## 3.设计与实现

---

# 算法设计

## 拓扑结构

限于服务器资源，我们至多收集到了4个服务器结点。由于服务器结点数目比较少，考虑采取环式的拓扑结构，尽量减少无效访问，相当于将各个结点的信息存在一个双向循环链表中。

## heart-beating

heart-beating算法用于检测一个结点是否故障。即周期性地向检测对象发送一条消息，若对方未回复，则认为对方故障。

在此项目中，daemon线程通过运行sendBeating线程与receiveBeating线程来隔一个特定的间隔就向检测对象发一条消息。根据拓扑结构可知，只需检测自己的下一个结点，并回复上一个结点即可。

同时我们设置了容错度与计时器的概念：一旦对方超过计时器设定的时长未回复消息，则认为出现一次错误。只有当连续出现大于容错度数目的错误，才会认为对方出现故障。

## gossip

gossip算法用于逐步向所有节点广播一条消息。

由于拓扑结构较为简单，服务器结点数目较少，我们仅设置传播量为1：即每次只向一个相邻的节点发送gossip消息，直至所有节点都已经收到新的push消息。

例如，当某个节点主动离开时，该节点会在离开前发送一条gossip消息，以一个传一个的形式告知所有节点；而当某个节点A被动离开时，负责检测其运行情况的节点B（也就是列表中A的上一个节点）会发现异常，然后向B的上一个节点C发送gossip消息，以此类推，不断进行gossip转发，直到所有节点都知晓A被动离开。

## join

将其中一个服务器设为introducer。每当有新的节点想要加入组时，都需要先对introducer发出申请，然后由introducer将这个新节点的信息广播给所有节点。

## 非功能性需求

- 可扩展性：组成员的数量不是固定的，可以随时添加。
- 低延迟：采用UDP方式通信，并尽可能使用多线程，降低时间延迟。
- 消息格式：gossip的消息格式如下：gossip sender type target。其中gossip标识这是一个gossip消息，sender存储发送这个gossip的起始节点的ip，target为发生离开/加入/故障的节点的ip，type则用于标识target发生的事件的类型。

## 与作业2的集成

在作业2中，我们主要实现了一个客户端对多个服务器的文件运输。但在作业2中，服务器的数量以及属性是固定的，无法动态地添加新的服务器，有服务器故障也无法及时的检测。而本次作业则在作业2的基础上，实现了组成员的服务，每次客户端发送请求时都可以知道现在有哪几个服务器是存活的，同时服务端也可以随时加入新的服务器。

## 丢包的处理

- 我们通过人为的方式模拟丢包，即每次发送消息前按照一定概率（3%/10%/30%）生成一个随机数，根据这个随机数来判断此次发送是否会“丢包”。
- 丢包的处理：
  - 在新节点发送消息时加入循环机制，若因丢包导致未成功加入则重复发送直至成功加入。
  - 在heart-beating中加入容错度机制，这样即使因为偶然的若干丢包未能接受回复消息，也不会误判对方发生故障。