

分布式系统 Task-1

21307289 刘森元

Prob.1 为什么有时候最大程度地实现透明性并不总是好的？

- 完全透明性是不可取的也是难以实现的，主要因为
 - 可能掩盖通信的性能问题
 - 完全隐藏网络和节点的失效是不可能的
 - 不能区分失效和性能变慢的节点
 - 不能确定系统失效之前的操作是什么
 - 完全的透明性可能牺牲性能，暴露系统分布特征
 - 保证复制节点与主节点的一致性需要时间（一致性问题）
 - 为了容错需要立即将内存修改的内容同步到磁盘上

Prob.2 可以通过应用多种技术来取得扩展性，都有哪些技术？

- 隐藏通信延迟
 - 异步通信技术
 - 设计分离的响应消息处理器
 - 将计算从服务器端移动到客户端
- 分布
- 复制
 - 复制文件服务器和数据库
 - Web站点进行镜像
 - Web缓存（在浏览器或者代理位置）
 - 文件缓存（在服务器和客户端）

Prob.3 中间件在分布式系统中扮演什么角色？

- 为系统集成提供通信设施

Prob.4 请从一些开源分布式软件如Hadoop、Ceph分布式文件系统、Apache Httpd、Spark等找出至少1处能够体现透明性的样例代码，并解释是何种类型的透明性

样例代码来自 Apache Hadoop 的分布式文件系统（HDFS）：

```
Configuration conf = new Configuration();
FileSystem fs = FileSystem.get(conf);
```

```
Path filePath = new Path("/path/to/file.txt");
FSDataInputStream inputStream = fs.open(filePath);
BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));

String line;
while ((line = reader.readLine()) != null) {
    // 操作文件内容
    System.out.println(line);
}

reader.close();
inputStream.close();
fs.close();
```

这段代码展示了 HDFS 的透明性，因为它隐藏了分布式文件系统的复杂性。开发者可以像操作本地文件一样，使用标准的 Java 文件操作 API（如 `BufferedReader`）来读取分布式文件系统中的文件。这种透明性使得开发者无需关心底层的分布式细节，可以更加专注于业务逻辑的实现。