本文的主要研究工作是如何在已知缺失区间和的情况下，对缺失数据进行填补，并针对复旦大学节能监管平台特定的数据特点，制定填补策略。 由于能耗数据的缺失场景复杂多样， 而填补算法的选择和填补策略的制定， 与缺失场景有着紧密的联系， 因此填补算法和策略有值得进一步研究的地方：

1） 当遇到缺失区间之间的间隔短的场景。 缺失区间的长度有长有短， 区间之间的间隔也不固定。 基于约束优化的缺失数据填补算法， 在计算时间序列相似度时， 需要用到缺失区间前一周和后一周的数据。 当遇到两个缺失区间之间的间隔很小的场景时， 那么用于计算相似度的数据就变得很少，影响相似时间序列的提取，从而影响填补效果。 此时，可以考虑两种解决方案：一是改进填补策略，针对该场景使用其他符合一定精度要求，且不需要缺失区间两侧数据作为预处理数据的填补算法来代替。 二是对填补算法进行改进，将两个间隔很短的缺失区间作为一个缺失区间进行处理，间隔部分再进行特殊处理。

2） 随着数据量的增大， 出现填补效率下降的情况。 能耗数据的采集是实时不断的，数据量在不断增大。 由于填补过程中需要多次检索数据，因此会导致填补效率下降。 缺失数据中有很多短期缺失数据，即缺失区间小于 24 小时的缺失数据，甚至有 2-3 小时以内的及短期的缺失区间。 如果此时填补速度很慢， 那么填补的性价比就变得很低。在这个情况下，可以改进填补算法的实现过程，减少对数据的检索。 另外可以考虑从并行计算方面进行效率上的改善。

先描述一下本文的情况。

1. 微服务架构方面的改进工作；
   1. 基于Java实现的微服务架构有些笨重，Golang
   2. 服务管理使用的是Docker-compose，灵活性不够，采用最新的调度k8s可能会更好；
   3. 云计算发展，可以考虑开发云服务，真正的实现BAAS(BlockChain as A Service)
2. 负载均衡算法方面的改进工作
   1. 随着服务范围的增大，服务节点的数量会相应提高，负载节点的

本文的主要工作是研究以太坊在服务过程中出现的性能低下和数据相关操作安全性低下的问题，针对以太坊在容器中的运行状态设计了高效的负载均衡算法，并根据私募股权管理应用场景下的具体业务设计和实现了基于反馈负载均衡策略的以太坊服务系统。本文的系统设计基于微服务思想，采用基于Java的Spring Cloud Alibaba解决方案。本文设计的相关方案以及系统实现仅适用以太坊在构建联盟链的应用场景中。为进一步提高该系统的服务效率，安全性等服务能力，可以考虑从系统架构以及负载均衡算法两个方面对系统进行优化：

1. 微服务架构方面的改进意见：

首先，本文以太坊服务系统实现是基于Spring Cloud Alibaba解决方案，该方案中各子系统的实现基于SpringBoot，而Java的运行需要JVM的支持，因此较为笨重，其开发效率与运行效率相对较低，可以考虑采用Google于2009年发布的Golang语言重构微服务系统中的各个子模块。Golang是编译型语言，它更适合多处理器的分布式系统，且其协程特性使得并发性能较Java更好。

其次，本文微服务系统中的服务管理采用的方式是Docker Compose，它在容器编排方面的灵活性不够。可以考虑引入Kubernetes提供容器的部署、编排和维护。Kubernetes会自动去新建、监控和重启服务，管理员可以加载一个微服务，让规划器来找到合适的位置，同时，Kubernetes也系统提升工具以及人性化方面，让用户能够方便的部署自己的应用。

最后，云计算时代已经到来，可以考虑将该服务系统进行上云处理，以实现真正的BaaS(BlockChain As A Service)。

1. 负载均衡算法优化：

本文主要考虑私募股权管理业务所面临的业务场景，因此，本负载均衡算法考虑容器的状态较少，换言之，本文所建立的容器状态向量空间的维度较低。因此，可以在深入研究以太坊客户端的运行特性和Docker容器的特性后，针对不同的应用场景设计更加高效的反馈负载均衡算法。随着云计算时代的到来，各种高效的负载均衡算法蓬勃发展，本文所设计的反馈负载均衡算法也应该与时俱进，且该领域潜力巨大，一定可以在本文所设计架构下提高以太坊服务能力。