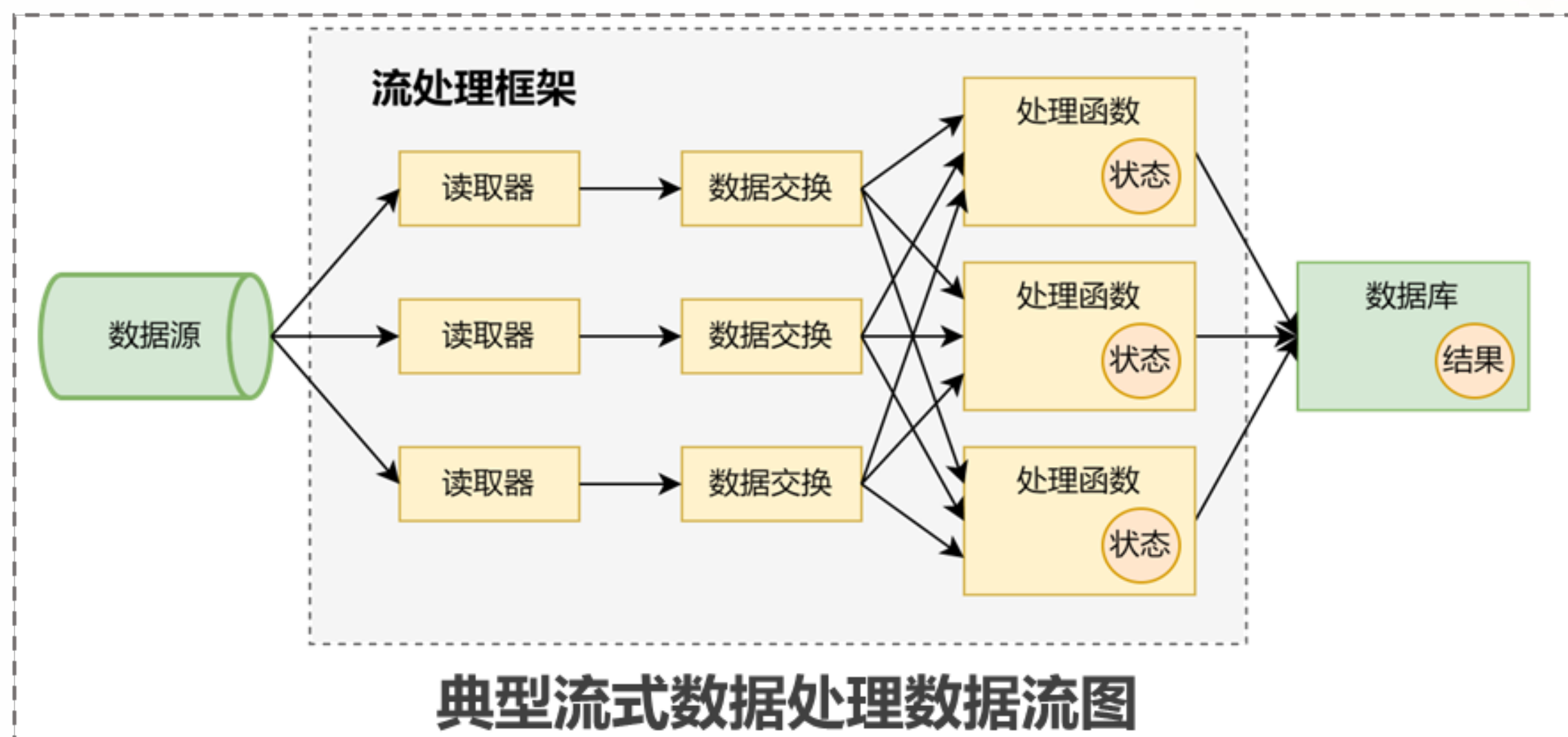


# 基于内存的高性能流式数据处理技术

研究团队：吴铭钲、王毅、曾鸿斌、秦政、许利杰、王伟  
软件工程技术研究开发中心

[xulijie09@otcaix.iscas.ac.cn](mailto:xulijie09@otcaix.iscas.ac.cn) [wangwei@otcaix.iscas.ac.cn](mailto:wangwei@otcaix.iscas.ac.cn)

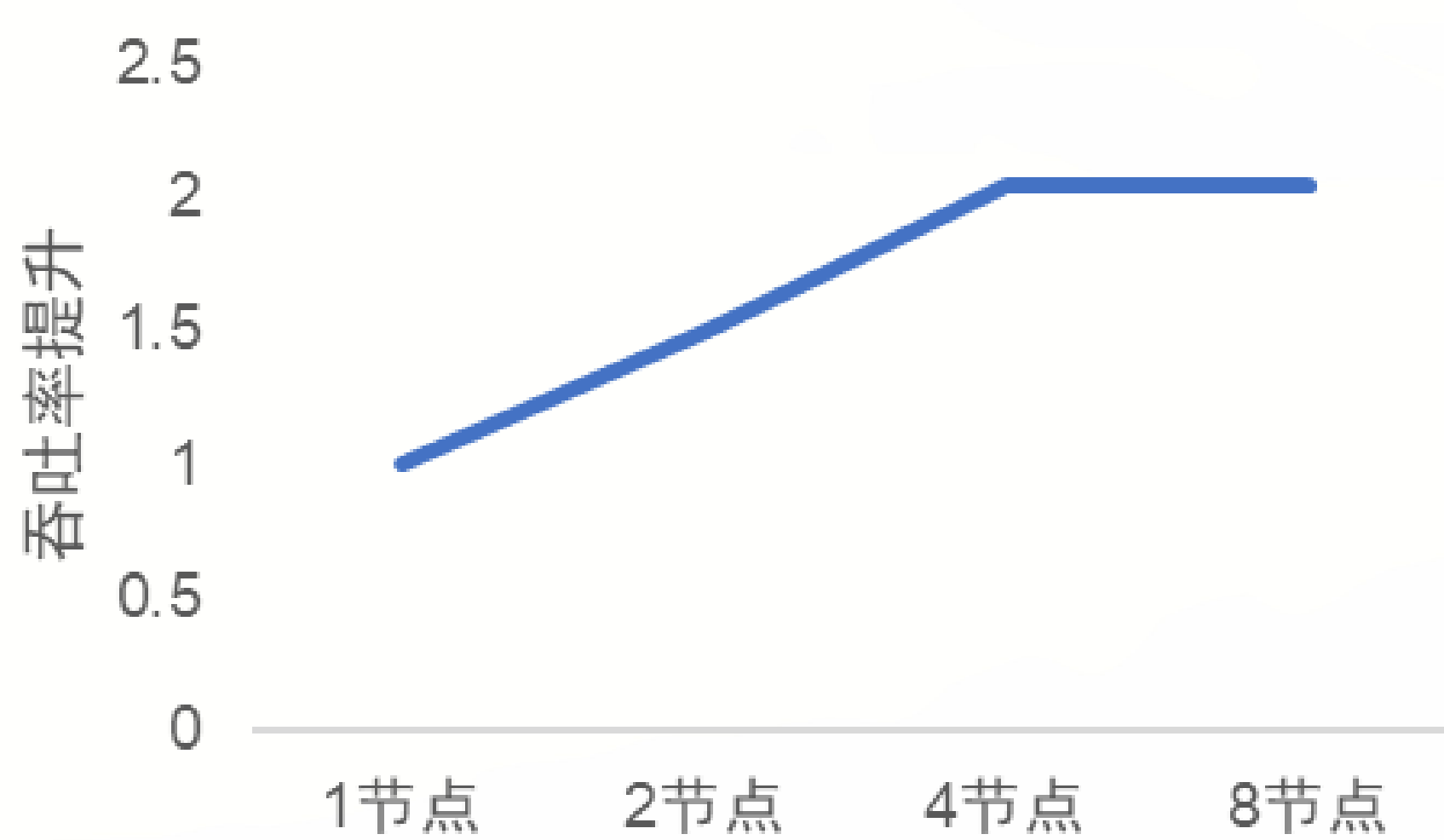
## 研究背景



### 流式分析水平可扩展性瓶颈

- 节点负载不均衡**：数据分配算法导致严重负载分配不均，实际生产中高负载节点往往比低负载节点额外承受一倍以上负载，形成瓶颈。
- 流数据时序耦合**：网络延迟差异和本地时钟偏差均会造成严重数据乱序，为保证正确性基于迟到数据进行结果更新会造成严重的吞吐下降。
- 节点间数据交换**：流分析中需要基于键值的数据交换以进行分析汇总，全量数据交换使网络带宽成为系统性能瓶颈，增加数据处理延迟。
- 数据链路低效率**：节点间数据传输链路策略受带宽和延迟制约，可能会选用低效网络链路，从而导致数据摄取的低吞吐和高延迟。

### 典型流处理框架可扩展性提升



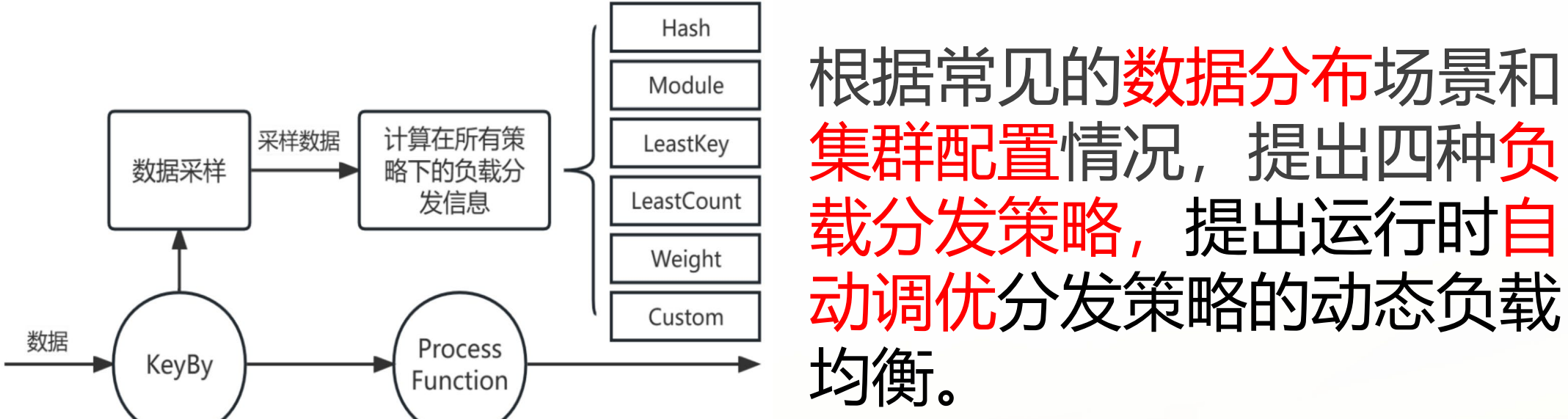
随着节点数增多，典型流处理框架性能并没有呈现线性增长，并且随节点增多甚至逐渐出现瓶颈，性能不再提升。

## 研究方法

### 节点负载不均衡

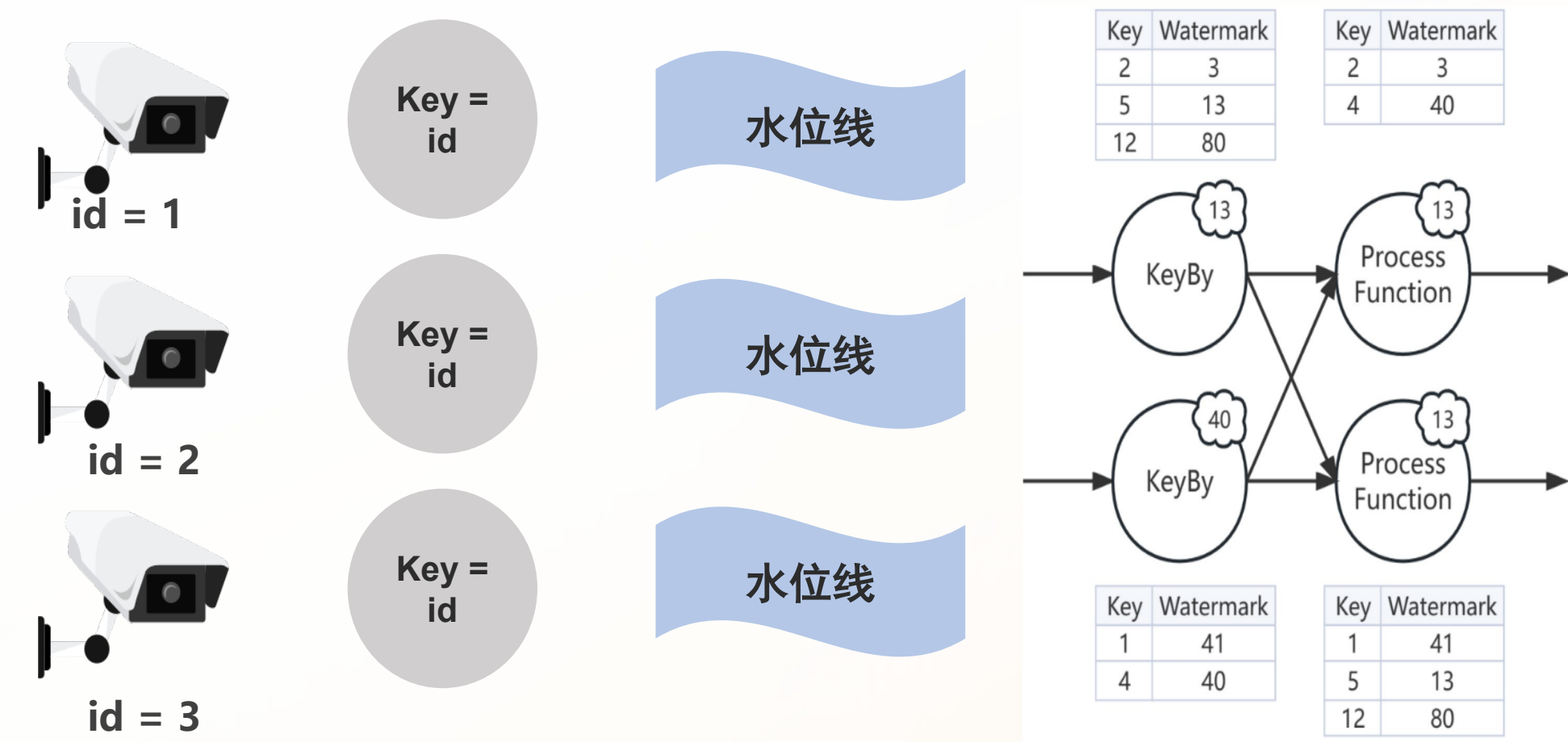
#### 负载分发策略与动态负载均衡

| Module    | LeastKey     | LeastCount  | Weight      |
|-----------|--------------|-------------|-------------|
| 对并行度取模    | 分配至Key负载最低节点 | 分配到历史最低负载节点 | 根据权重分配      |
| Key均匀分布场景 | 数据均匀分布场景     | 数据不均匀分布场景   | 节点间硬件配置不同场景 |



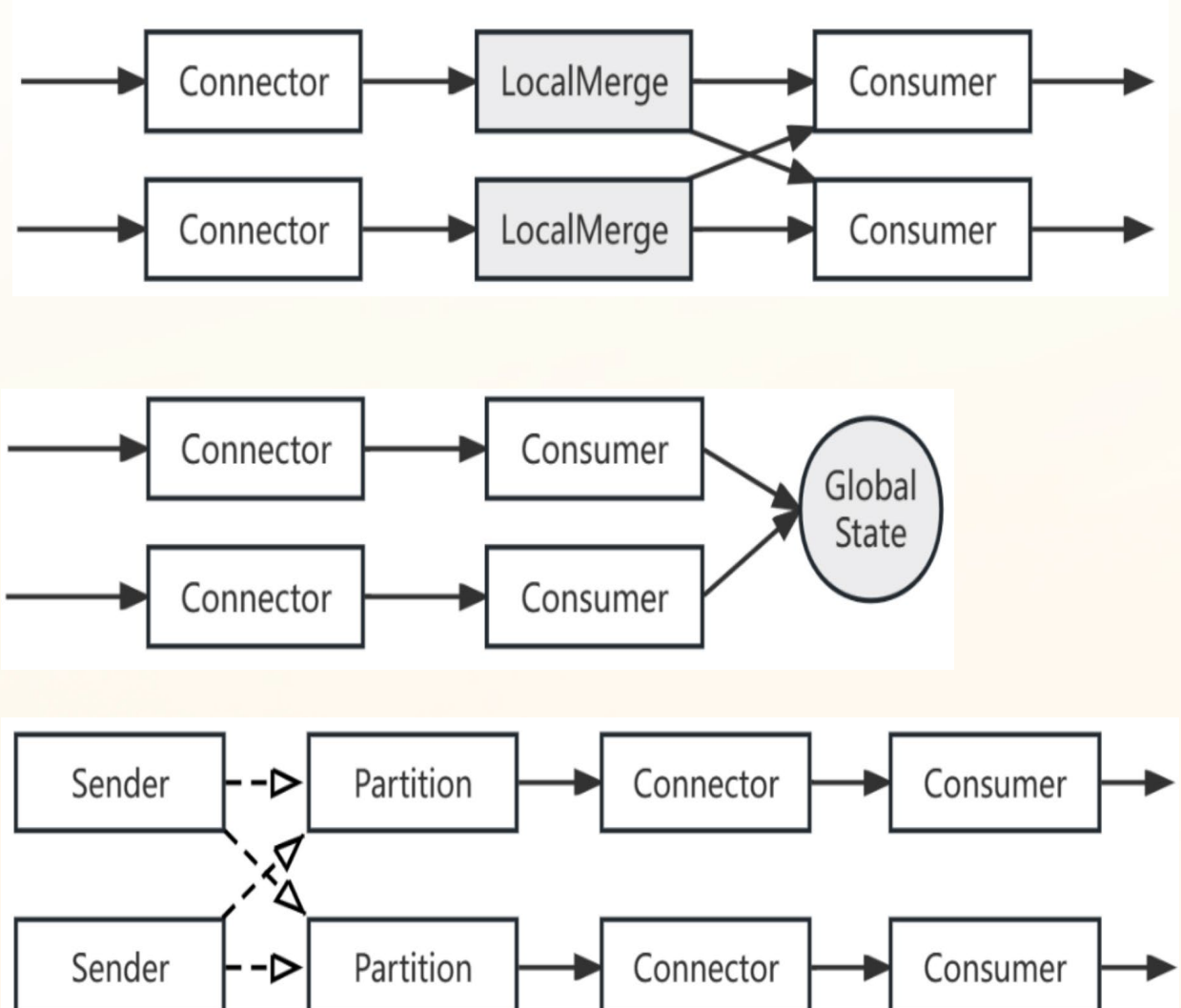
### 流数据时序耦合

#### 主键级水位线时序保证



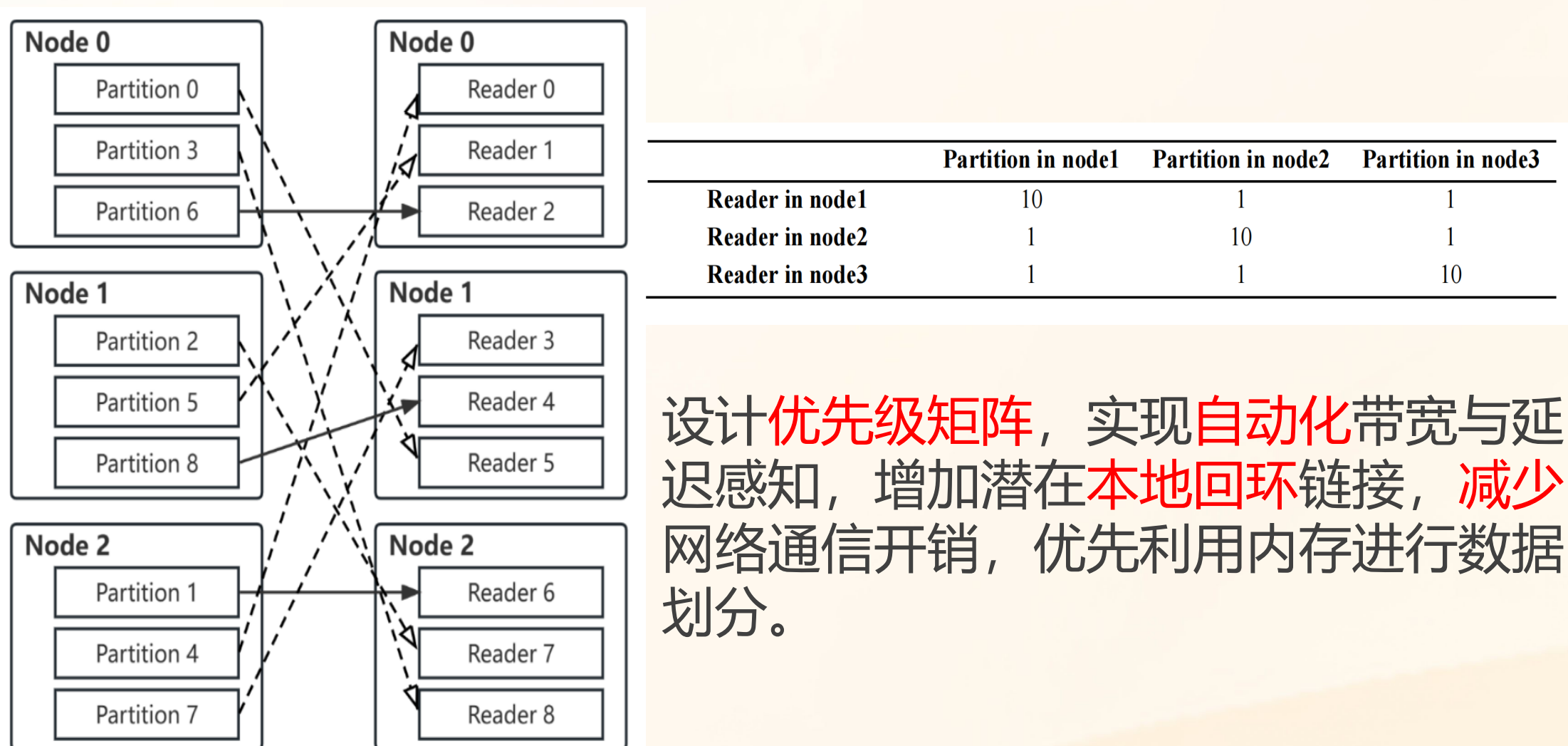
### 节点间数据交换

#### 基于键值的数据交换策略



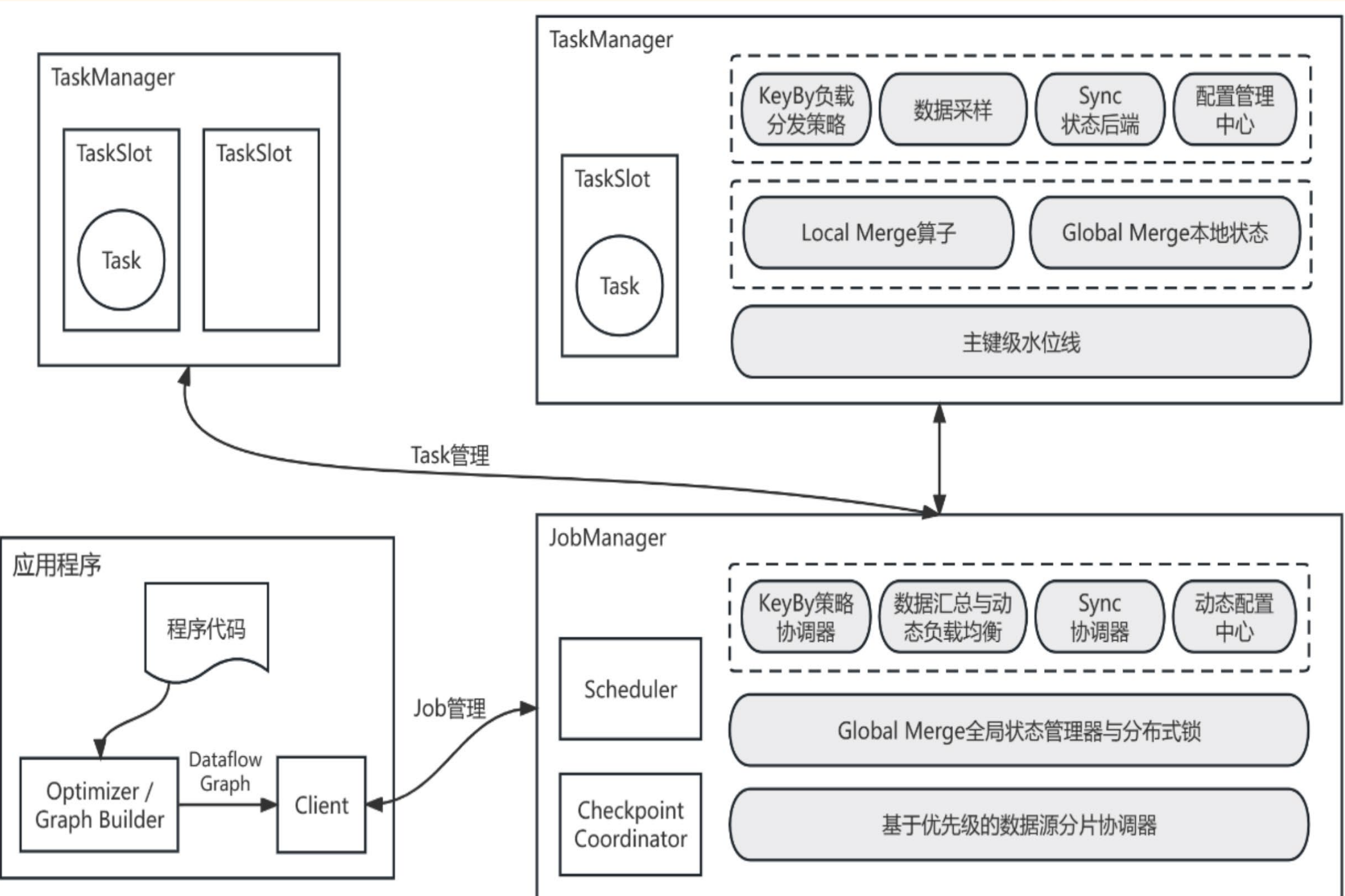
### 数据链路低效率

#### 基于优先级的数据源分片策略

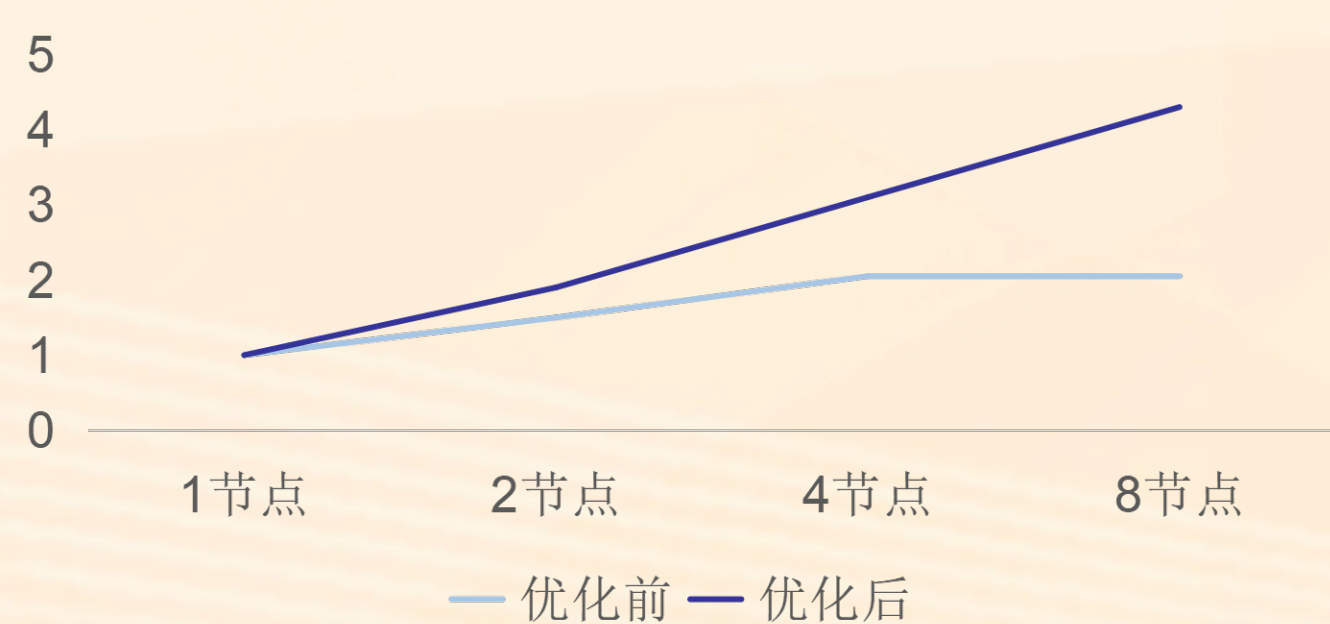


## 系统实现

### 基于Flink框架的系统实现

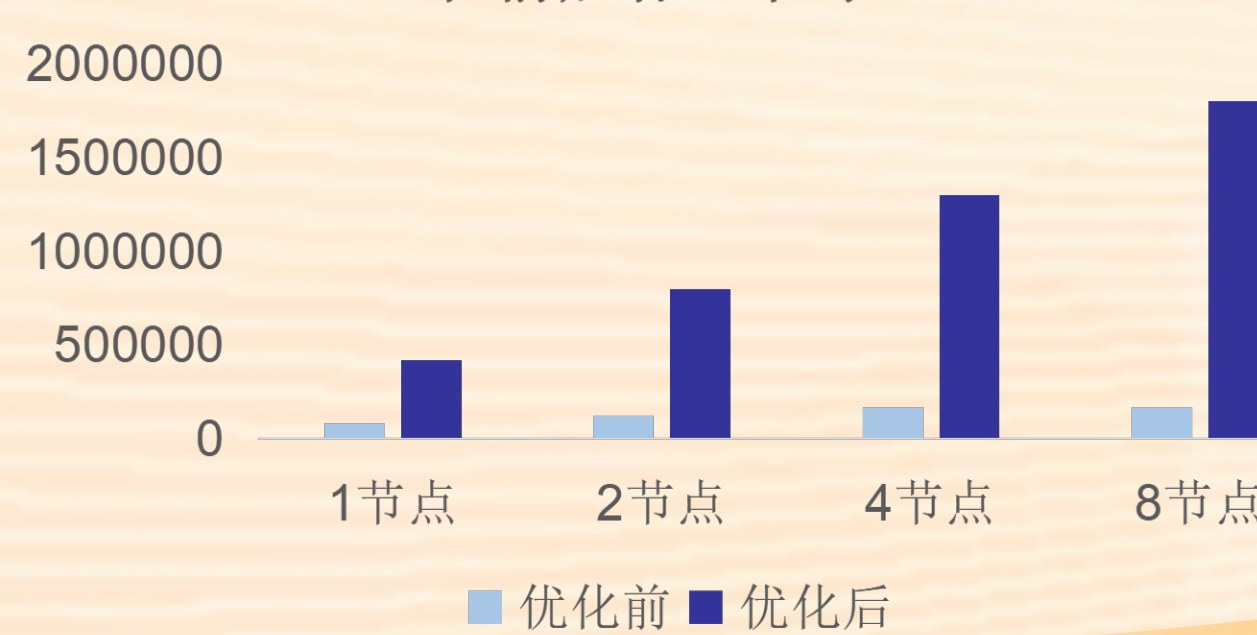


### 基于内存的高性能流式数据处理技术优化前后可扩展性对比



优化后，系统性能随节点数增加呈亚线性增长趋势

### 基于内存的高性能流式数据处理技术优化前后吞吐量对比



优化后，系统性能在八节点集群吞吐率达180万条每秒，相较于优化前提升十倍以上。