



탄력성, 고가용성, 고성능, 데이터 일관성을 고려한 Distributed Processing Solution을 개발하려고 하는데, 클래스 다이어그램으로 간략하게 구조 설계 예시를 보여줘

Distributed Processing Solution을 위한 클래스 다이어그램 예시는 다음과 같이 설계할 수 있습니다.

주요 클래스 및 역할

1. ClusterManager

- 클러스터 전체 상태 관리 및 리더 선출, 노드 상태 모니터링 담당
- 속성: nodeList, leaderNode, clusterState
- 메서드: electLeader(), monitorNodes(), recoverNode()

2. Node

- 분산 시스템 내 각 노드를 나타냄
- 속성: nodeId, status, resourceUsage, isLeader
- 메서드: sendHeartbeat(), processTask(), replicateData()

3. TaskScheduler

- 분산 작업 스케줄링 및 부하 분산 담당
- 속성: taskQueue, activeTasks
- 메서드: scheduleTask(), balanceLoad()

4. DataReplicator

- 데이터 복제 및 일관성 유지 메커니즘 담당
- 속성: replicationFactor, consistencyLevel
- 메서드: replicate(), syncData(), resolveConflicts()

5. MessageQueue

- 비동기 메시지 전달, 이벤트 처리 담당 (Kafka, RabbitMQ 등)
- 속성: queueName, messages
- 메서드: enqueue(), dequeue(), acknowledge()

6. MonitoringService

- 시스템 상태 모니터링 및 알림 담당

- 속성: metrics, alerts
- 메서드: collectMetrics(), sendAlert(), analyzeLogs()

클래스 관계

- ClusterManager는 여러 Node를 관리하며 리더 노드를 elect한다(집합 관계).
- Node는 TaskScheduler를 통해 분산 작업을 할당받고 실행한다(연관 관계).
- DataReplicator는 Node 간 데이터 일관성을 위해 동작한다(의존 관계).
- MessageQueue는 TaskScheduler와 Node 간 비동기 통신을 지원한다(연관 관계).
- MonitoringService는 ClusterManager와 Node의 상태를 지속적으로 수집 및 분석한다(의존 관계).

이 구조는 탄력적 확장과 고가용성 확보를 위해 클러스터 관리와 노드 상태 감시를 중심으로 하며, 고성능을 위해 작업 스케줄링과 비동기 메시징을 결합한다. 데이터 일관성은 복제 및 동기화 클래스에서 강화한다.

간략한 UML 클래스 다이어그램 형태로 이 설계가 표현될 수 있습니다.

이 구조를 바탕으로 구체적인 시스템 요구사항에 맞추어 속성, 메서드, 관계를 확장 및 상세화할 수 있습니다.



1. <https://www.edrawsoft.com/kr/diagram-tutorial/learn-about-class-diagram.html>
2. <https://nalssen.tistory.com/entry/프로젝트-설계2-시퀀스-다이어그램-클래스-다이어그램>
3. <https://chiba.tistory.com/81>
4. <https://sabarada.tistory.com/72>
5. <https://blog.naver.com/he1000/220725205602>
6. <https://translate.google.com/translate?u=https%3A%2F%2Fwww.quora.com%2FWhat-is-the-best-UML-diagram-for-distributed-systems&hl=ko&sl=en&tl=ko&client=srp>
7. <https://velog.io/@yooha9621/UML다이어그램UML다이어그램-종류1>
8. <https://miro.com/ko/diagramming/what-is-a-uml-diagram/>
9. <https://www.promleeblog.com/blog/post/64-uml-class-diagram>
10. <https://www.processon.io/ko/blog/uml-class-diagram-methods>