



№3 Дәріс. Үлестірілген жүйелердің архитектурасы мен компоненттері

Дәріскер: Даркенбаев Д.К. PhD, қауымдастырылған
профессор

Дәріс мақсаты

Архитектуралық модельдер

Үлестірілген жүйелердің архитектуралық модельдерін түсіндіру

Негізгі компоненттер

Негізгі компоненттерін және олардың өзара әрекеттесуін сипаттау

Жұмыс принциптері

Үлестірілген жүйелердің жұмыс істеу принциптерін, артықшылықтары мен қиындықтарын талдау

Кілт сөздер

- Үлестірілген жүйе
- Архитектура
- Түйін (node)
- Сервер
- Клиент
- Репликация
- Синхрондау
- Ортақ ресурстар
- RPC
- Middleware
- P2P
- Кластер
- Бұлттық архитектура
- Fault tolerance

Дәріс жоспары

О1

Үлестірілген жүйелерге кіріспе

О2

Үлестірілген жүйе архитектуралары

- Клиент-сервер архитектурасы
- Peer-to-peer архитектурасы
- N көпдеңгейлі (multi-tier) архитектура
- Микросервис архитектурасы
- Бұлттық және контейнерлік архитектуралар

О3

Үлестірілген жүйелердің негізгі компоненттері

- Түйіндер (Nodes)
- Байланыс модулі (Communication subsystem)
- Middleware
- Деректерді сақтау және репликация модулі
- Атаулар мен каталог қызметі (Naming service)
- Тапсырмаларды жоспарлау (Scheduler)

О4

Өнімділік, масштабталу және fault-tolerance

О5

Қорытынды

1. Үлестірілген жүйелерге кіріспе

Үлестірілген жүйе – өзара байланысқан бірнеше компьютерлер немесе серверлер жиыны, олар бір мақсатқа жұмыс істейді және пайдаланушыға бірыңғай жүйе ретінде көрінеді.

Мақсаты:

- Ресурстарды ортақ пайдалану
- Өнімділікті арттыру
- Сенімділік пен қолжетімділік
- Үлкен көлемді деректер мен операцияларды өңдеу





1. Үлестірілген жүйе архитектуралары



1. Клиент — Сервер архитектурасы

Классикалық модель: Client → Network → Server

Жұмыс принципі

- Клиент сұраныс жібереді
- Сервер өңдейді және жауап қайтарады

Түрлері:

- 2-tier (клиент–сервер)
- 3-tier (қосымша сервер + DB)
- n-tier (микросервис, cloud)

Қолдану мысалдары:

Web жүйелер, банктік жүйелер

1. Peer-to-Peer (P2P)

Әр түйін бір уақытта сервер және клиент:

Node ↔ Node ↔ Node

Қолдану:

Torrent, Blockchain, IPFS



1. Кластерлік архитектура

Түйіндер тобы бір жүйе ретінде жұмыс істейді

Мақсат:

HPC (supercomputing), параллель есеп

□ Node1 + Node2
+ Node3 ... = 1
жүйе

An isometric illustration of a microservices architecture. It features a central stack of seven server-like blocks in shades of green and blue, topped with a white cube containing a circular hole. This stack sits on a white, rounded square base. Surrounding the base are several smaller white cubes, some with green details, connected by thin white lines. The background is a dark blue gradient with stylized green hills and a single green cloud.

1. Микросервис архитектурасы

Қызметтер тәуелсіз, API арқылы байланысады

Service1 — API — Service2 — API — DB

Пайдасы:

- Масштабталу
- Fault isolation

Мысал:

Google, Netflix, Amazon



1. Бұлттық және контейнерлік архитектура



Контейнерлер

Docker / Kubernetes



Бұлттық жүйелер

AWS, Azure, GCP

3. Үлестірілген жүйелердің негізгі компоненттері

1	Түйіндер (Nodes) Компьютерлер, серверлер, виртуал машиналар, контейнерлер.
2	Байланыс модулі Түйіндер арасындағы ақпарат алмасу: - TCP/IP - HTTP/HTTPS - RPC (Remote Procedure Call) - gRPC, REST API
3	Middleware (аралық бағдарлама) Жүйе компоненттерін байланыстырады: - Message brokers (Kafka, RabbitMQ) - ORB, CORBA - Service Mesh (Istio) Функциялары: - Авторизация - Кэштеу - Load balancing
4	Деректерді сақтау және репликация Репликация: - Master-Slave - Multi-Master - Eventually Consistent System Мысалдар: - Cassandra - MongoDB - Hadoop HDFS
5	Naming & Directory service Түйіндерді, қызметтерді табу DNS, Consul, Zookeeper
6	Scheduler (тапсырмаларды жоспарлау) Жұмыстарды бөледі Мысалдар: - Kubernetes scheduler - Hadoop YARN

4. Өнімділік және Fault Tolerance

Үлестірілген жүйелердің басты талаптары:

Критерий	Түсінігі
Масштабталу	Жүйе кеңейе алуы
Қолжетімділік	Жүйе әрқашан жұмысқа дайын
Fault Tolerance	Қателерге төзімділік
Консистенттілік	Деректердің дұрыстығы

CAP теоремасы



❏ Үлестірілген жүйе бір уақытта тек екеуін ғана толық қамтамасыз етеді.

Қорытынды

Үлестірілген жүйелер:

Жоғары өнімділік

Жоғары өнімділікті қамтамасыз етеді

Үлкен деректер

Үлкен деректермен жұмыс істейді

Сенімді инфрақұрылым

Масштабталатын және сенімді инфрақұрылым береді

Архитектура таңдау — жүйе мақсатына байланысты.

Бақылау сұрақтары

- 1 Үлестірілген жүйе деген не?
- 2 Клиент-сервер және P2P айырмашылығы?
- 3 Middleware рөлі қандай?
- 4 Репликация түрлері?
- 5 CAP теоремасы не дейді?
- 6 Микросервис пен монолит айырмашылығы?
- 7 Кластер қандай мақсатта қолданылады?

Пайдаланылған әдебиеттер:

1. Tanenbaum, A. — Distributed Systems: Principles and Paradigms
2. Coulouris, G. — Distributed Systems Concepts and Design
3. Kai Hwang — Distributed and Cloud Computing
4. Rajkumar Buyya — High Performance Cluster Computing
5. IEEE, ACM Digital Library articles
6. Kubernetes, Docker, AWS Documentation