

№7 Дәріс. Үлестірілген жүйелердегі қателер және ақауларға төзімділік

Дәріс мақсаты:

- Үлестірілген жүйелердегі қателердің түрлері мен себептерін түсіндіру;
- Ақауларға төзімділік (fault tolerance) ұғымын түсіндіру және оның маңызын көрсету;
- Қателерді анықтау, алдын алу және қалпына келтіру әдістерімен таныстыру;
- Репликация, журналдау (logging), бақылау (monitoring) және консенсус механизмдерін қарастыру.

Кілт сөздер:

Ақауларға төзімділік, таратылған жүйенің істен шығуы, істен шығу, желінің бөлінуі, Византия ақауы, репликация, бақылау пункті, консенсус, Paxos, Raft, резервтеу.

Дәріс жоспары

1. Үлестірілген жүйелердегі қателер түсінігі
2. Қателердің негізгі түрлері
3. Ақауларға төзімділік ұғымы
4. Ақауларды анықтау және қалпына келтіру тәсілдері
5. Репликация стратегиялары
6. Checkpointing және журналдау
7. Консенсус алгоритмдері: Paxos, Raft
8. Қорытынды

Кіріспе

1. Үлестірілген жүйелердегі қателер

Үлестірілген жүйелерде көптеген түйіндер, серверлер, желілік құрылғылар мен бағдарламалар өзара әрекеттеседі. Сондықтан қателер болуы — қалыпты жағдай.

Қате (failure) — жүйенің күтілген қызметін орындай алмауы.

Ақау (fault) — қателікке себеп болатын жағдай.

Қателік көрінісі (error) — ақаудың жүйе жұмысына әсері.

2. Қателердің негізгі түрлері

Жүйедегі қателердің типтері:

Қате түрі	Түсінігі
Crash Failure	Түйін толық істен шығады (күтпеген тоқтау)
Omission Failure	Хабарлама жоғалуы
Timing Failure	Уақыт кешігу, жауап кешігу
Network Partition	Желі бөліктерге бөлінеді
Byzantine Failure	Түйін қате немесе зиянды әрекет жасайды
Performance Failure	Жүйе баяулайды немесе жұмыс көлемін көтере алмайды

Byzantine Fault (күрделі қате)

Түйін дұрыс емес немесе қасақана жалған ақпарат береді. Бұл ең ауыр қате түрі.

Мысал: блокчейн желісіндегі зиянды түйіндер.

3.Ақауларға төзімділік (Fault Tolerance)

Анықтама:

Жүйе қателер болған жағдайда да жұмысын жалғастыра алуы.

Негізгі принциптер:

- Redundancy — қосымша ресурстар (резервтік серверлер)
- Replication — деректердің көшірмелерін сақтау
- Failover — жұмыс істемей қалған түйінді басқамен ауыстыру
- Self-healing — автоматты қалпына келу

Мақсаты:

- Жүйені тоқтатпау
- Деректерді жоғалтпау
- Қызмет көрсетудің үздіксіздігі

4.Ақауларды анықтау және қалпына келтіру тәсілдері

Әдіс	Мәні
Failure detection	Ping, heartbeat, monitoring
Retry / Timeout	Қайта жіберу, уақыт шекарасы
Failover	Басқа түйінге ауысу
Rollback	Соңғы сақталған күйге оралу
Replication	Деректер көшірмесін қолдану
Graceful degradation	Мүмкіндікті шектеп, жұмысын жалғастыру

5.Репликация стратегиялары

Репликация түрлері:

Түрі	Мәні
Primary–backup	Негізгі сервер және сақтық көшірмелер
Active–active	Барлық репликалар жұмыс істейді
Synchronous replication	Барлық репликаға бір уақытта жазу
Asynchronous replication	Жазу кейін синхрондалады

Synchronous + күшті консистенттілік,

Async + жоғары жылдамдық / eventual consistency

Қолдану мысалдары:

- Google Cloud Spanner — synchronous replication
- Amazon S3, Cassandra — eventual consistency

6.Checkpointing және журналдау

Техника	Анықтамасы
Checkpointing	Жүйенің күйін үнемі сақтау

Техника	Анықтамасы
Logging (журналдау)	Барлық операциялар журналға жазылады

Ақау болғанда соңғы checkpoint журналдағы операциялар жүйе қалпына келтіріледі.

7.Консенсус алгоритмдері: Paxos және Raft

Үлестірілген жүйеде барлық түйіндер ортақ шешім қабылдауы тиіс.

Негізгі консенсус міндеттері:

- Түйіндер келісімге келуі
- Деректердің дұрыс және бірдей болуы
- Қате түйіндер болса да нәтиже алу

Танымал алгоритмдер

Алгоритм	Ерекшелігі
Paxos	Классикалық, теориялық, күрделі
Raft	Түсінуге оңай, қазіргі жүйелерде жиі қолданылады

Қолдану:

- Etcd (Kubernetes)
- Consul
- Google Chubby
- Apache ZooKeeper

Қорытынды

Үлестірілген жүйелерде қателер — табиғи құбылыс.

Тиімді fault tolerance стратегиясы жүйені:

- сенімді
- өнімді
- үздіксіз жұмыс істейтін

етеді.

Негізгі принцип:

Қате емес — қателікті ескермеу қауіпті.

Бақылау сұрақтары

1. Ақауларға төзімділік деген не?
2. Үлестірілген жүйелерде кездесетін негізгі қателер түрлері қандай?
3. Crash failure және Byzantine failure айырмашылығы неде?
4. Репликация деген не? Түрлерін атаңыз.
5. Checkpointing не үшін қолданылады?
6. Paxos және Raft алгоритмдері қай мақсатта қолданылады?

Пайдаланылған әдебиеттер

1. Tanenbaum A., Steen M. *Distributed Systems: Principles and Paradigms*
2. Coulouris G. *Distributed Systems: Concepts and Design*
3. Leslie Lamport — *Paxos Made Simple*
4. Ongaro, Ousterhout — *Raft Consensus Algorithm*
5. Google Cloud Spanner Technical Paper
6. Apache ZooKeeper Documentation