

№8 Дәріс. Репликация және келісім (consensus) алгоритмдері (Paxos, Raft)

Дәріскер: Даркенбаев Д.К. PhD, қауымдастырылған
профессор



Дәрістің мақсаты

Үлестірілген жүйелерде репликация ұғымын, оның артықшылықтарын және деректердің тұтастығын қамтамасыз ету әдістерін түсіндіру. Paxos және Raft алгоритмдерінің жұмыс принциптері мен айырмашылықтарын меңгерту.

Кілт сөздер

Репликация, consensus, Paxos, Raft, fault tolerance, leader election, лог репликациясы, distributed systems, fail-stop, majority voting.

Кіріспе

Заманауи үлестірілген жүйелер (бұлттық қызметтер, банк жүйелері, әлеуметтік желілер) үздіксіз жұмыс істеп, деректерді жоғалтпауға тиіс. Сервер істен шықса да ақпарат жоғалмауы керек. Мұны қамтамасыз етудің негізгі тәсілі – репликация және келісім (consensus) алгоритмдері.

1. Репликацияның мәні

❏ **Анықтама.** Репликация – деректердің бірнеше серверде (түйіндерде) көшірмесін сақтау процесі. Мақсаты – жоғары қолжетімділік, деректер сенімділігі және жылдамдық.

Репликацияның артықшылықтары:

Қолжетімділік	Сервер тоқтағанда да жүйе жұмыс істейді
Жылдамдық	Жақын орналасқан репликадан оқу жылдамырақ
Қауіпсіздік	Деректер жоғалу қаупі төмендейді

Репликация түрлері:

Синхронды	Барлық реплика деректі бір уақытта жазады
Асинхронды	Негізгі түйін жазып, кейін басқаларына таратады
Гибридті	Екі тәсілдің комбинациясы

1. Consensus (келісім) деген не?

📄 **Анықтама.** Consensus — бірнеше тәуелсіз түйіннің ортақ мәнге (шешімге) келісуі.

Қойылатын шарттар:

- **Agreement** — барлық дұрыс түйіндер бір мәнге келісуі керек
- **Validity** — шешім нақты ұсынылған мән болуы керек
- **Termination** — алгоритм шексіз күтуге болмайды (қорытынды шығару тиіс)
- **Fault tolerance** — ақауларға төзімділік

Consensus TCP/IP желілерінде қиын, себебі:

- Пакеттер жоғалады
- Желі кідірісі болады
- Серверлер істен шығады
- Кейбір түйіндер дұрыс емес ақпарат беруі мүмкін (Byzantine fault)

1. Paxos алгоритмі

❏ **Анықтама.** Paxos — Лесли Лэмпорт ұсынған консенсус алгоритмі, үлестірілген жүйелердегі стандарт механизм.

Компоненттер:

- **Proposers** — мән ұсынады
- **Acceptors** — ұсынысты қабылдайды
- **Learners** — нәтижені оқиды

Жұмыс кезеңдері:

Prepare	Proposer «дайынсың ба?» деп acceptors-қа хабар жібереді
Promise	Acceptors тек үлкен нөмірлі ұсыныстарды қолдайтынын айтады
Accept	Proposer ұсыныс мәнін жібереді
Accepted	Acceptors қабылдап, нәтиже жүйеге беріледі

Артықшылықтары

- Ақауларға өте төзімді
- Академиялық түрде дәлелденген

Кемшіліктері

- Түсіну қиын
- Іске асыру күрделі
- Түсініксіз инженерлік құжаттама

Google Chubby, ZooKeeper (Zab) Paxos идеяларына негізделген.

1. Raft алгоритмі

📄 **Анықтама** Raft — консенсус алгоритмі, Paxos-қа альтернатива. 2014 жылы Диего Онгаро және Джон Оустерхаут әзірлеген. Қарапайым әрі түсінікті болу үшін жазылған.

Raft компоненттері

- **Leader** — жазуларды басқарады
- **Followers** — көшірме сақтайды
- **Candidates** — лидер сайлауға қатысады

Жұмыс принципі:

Leader Election	Лидер таңдау
Log Replication	Командаларды барлық түйіндерге тарату
Safety	Келісім мен деректер тұтастығы

Артықшылықтары

- Түсінікті архитектура
- Код жазу оңай
- Қалыпты инженерлік тілде сипатталған

Қолданылатын жүйелер

- Etcd (Kubernetes)
- Consul
- TiDB
- RethinkDB

1. Paxos және Raft салыстыру

Түсініктілігі	Қиын	Жеңіл түсіндіріледі
Жобалауда қолдану	Сирек	Кеңінен қолданылады
Құжаттама	Лэмпорт қағаздары	Толық, инженерлерге арналған
Лидерлік	Әр түрлі нұсқалар	Қатаң leader-based
Product-ready	Google жүйелері	Kubernetes экожүйесі

1. Мысал: Кворум (quorum) тұжырымдамасы

Consensus жүйелерінде шешім қабылдау үшін кворум қажет. Мысалы, 5 түйін болса:

- **3 түйін келіссе** — шешім қабылданады
- 2 түйін істен шықса да жүйе жұмыс істейді



1. Жүйенің ақауларға төзімділігі

Fail-stop fault model

Түйін тоқтап қалады → бірақ бұзылмайды.

Byzantine fault model

Түйін қате, жалған ақпарат береді → ең қиын модель (blockchain осылай жұмыс істейді)

Consensus алгоритмдері біз қарастырған жағдайда **fail-stop моделін** қолдайды.



Қорытынды

Репликация және келісім — үлестірілген жүйелердің негізі.

Paxos

Теориялық, дәлелденген, бірақ күрделі

Raft

Индустрияда көбірек қолданылатын, түсінікті және практикалық

Бақылау сұрақтары

1. Репликация деген не және не үшін қажет?
2. Consensus алгоритмдері қандай мәселелерді шешеді?
3. Paxos алгоритмінің негізгі кезеңдері қандай?
4. Raft алгоритміндегі лидер сайлау процесін түсіндіріңіз.
5. Синхронды және асинхронды репликация айырмашылығы қандай?
6. Кворум деген не және не үшін керек?
7. Paxos және Raft айырмашылықтары қандай?

Пайдаланылған әдебиеттер

1. Lamport, L. "Paxos Made Simple", 2001.
2. Ongaro, D., Ousterhout, J. "In Search of an Understandable Consensus Algorithm", 2014.
3. Tanenbaum, A. "Distributed Systems: Principles and Paradigms".
4. Martin Kleppmann — Designing Data-Intensive Applications.
5. Google Chubby technical paper, 2006.
6. etcd & Raft documentation — CNCF, 2020.

