

Juuso Lempeä

NoSQL-tietokannat MMO-pelinkehityksessä

Tietotekniikan kandidaatintutkielma

27. toukokuuta 2024

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Juuso Lempeä

Yhteystiedot: juuso.p.lempea@student.jyu.fi

Ohjaaja: Annemari Auvinen

Työn nimi: NoSQL-tietokannat MMO-pelinkehityksessä

Title in English: NoSQL databases in MMO game development

Työ: Kandidaatintutkielma

Opintosuunta: Tietotekniikka

Sivumäärä: 24+0

Tiivistelmä: Tiedonhallintaa ja tietokantoja ei tyypillisesti pidetä merkittävänä tai edes tarpeellisina perinteisessä videopelinkehityksessä, mutta massiivisissa monen pelaajan verkkopeleissä eli MMO-peleissä pelaajat jakavat yhteisen pysyvän pelimaailman, jonka ylläpitoon tarvitaan tietokannan hallintajärjestelmää. Tässä kirjallisuuskatsauksena toteutetussa kandidaatintutkielmassa pyrittiin selvittämään, miten NoSQL-tietokantoja voidaan hyödyntää MMO-pelinkehityksessä, ja olisivatko ne perinteisiä relaatiotietokantoja parempi vaihtoehto MMO-arkkitehtuurissa. Tutkielmassa käsitellyn kirjallisuuden perusteella NoSQL-tietokannat vaikuttavat lupaavilta MMO-pelinkehityksessä.

Avainsanat: MMO, pelinkehitys, tietokannat, tiedonhallinta, NoSQL, pilvipalvelut

Abstract: Data management and databases are not typically considered significant or even necessary in traditional video game development, but in Massively Multiplayer Online Games (MMOGs) players share a persistent game world, which requires the use of a database management system. This literature review aimed to discover how NoSQL databases can be utilised in MMOG development and whether they would be a better alternative than traditional relational databases for MMOG architecture. The literature reviewed for this Bachelor's thesis indicates that NoSQL databases seem promising in MMOG development.

Keywords: MMO, game development, databases, data management, NoSQL, cloud computing

Termiluettelo

ACID	Oikeellisuusmalli, joka takaa tietokannan transaktioiden olevan atomisia, eheitä, eristyneitä ja pysyviä.
BASE	Oikeellisuusmalli, joka pyrkii takaamaan tietokannalle mahdollisimman hyvän saatavuuden CAP-teoreeman mukaisesti eheyden kustannuksella.
CAP	Teoreema, jonka mukaan hajautettu verkkopalvelu voi taata vain kaksi seuraavista ominaisuuksista: eheys, saatavuus ja pirstoutumisen sietokyky.
Eheys (ACID)	Takaa tietokannan säilyvän tietokannan sisäisiä sääntöjä ja rajoitteita noudattavassa tilassa transaktion suorituksessa.
Eheys (CAP)	Kaikissa verkon solmuissa on aina saatavilla yhtenäinen uusin versio datasta. Poikkeaa ACID-eheydestä.
MMO-peli	Peli, jossa suuri määrä pelaajia vuorovaikuttaa toistensa kanssa jaetussa pysyvässä pelimaailmassa.
NoSQL-tietokanta	Tietokanta, joka ei noudata relaatiomallia.
Pilvipalvelu	Palvelu, joka toteuttaa laskentaa tai tiedonhallintaa muussa kuin käyttäjän laitteessa.
Pysyvä pelimaailma	Pelimaailma, jonka ylläpitoa jatketaan pelaajien vuorovaikutuksesta riippumatta.
Rakenteeton data	Dataa, jolla ei ole metatietojen avulla määritettyä rakennetta. Esim. teksti- ja pdf-tiedostot, kuvat, videot, ääni, verkkosivut.
Relaatiomalli	Tiedonhallintamalli, jossa tieto esitetään riveistä ja sarakkeista koostuvina toisiinsa yhdistyvinä tauluina.

Sisällys

1	JOHDANTO	1
2	TIETOKANNAT	3
	2.1 Relaatiotietokannat	3
	2.2 CAP-teoreema ja BASE-oikeellisuusmalli	4
	2.3 NoSQL-tietokannat	5
3	MMO-PELIEN ANATOMIA	7
	3.1 Pysyvät ja ei-pysyvät pelit	7
	3.2 Asiakas-palvelin -arkkitehtuuri	8
4	NOSQL-TIETOKANNAT MMO-PELINKEHITYKSESSÄ	10
	4.1 Pilvipalvelut ja mikropalveluarkkitehtuuri	10
	4.2 Esimerkkejä NoSQL-tietokantoja hyödyntävistä peleistä	11
5	YHTEENVETO	14
	LÄHTEET	16

1 Johdanto

Tietokannan hallintajärjestelmät (engl. *database management system, DBMS*) ovat olleet keskeinen osa ohjelmistokehitystä vuosikymmenien ajan. Tietokoneohjelmistojen mittakaavan ja käyttäjämäärien kasvaessa käsiteltävän ja tallennettavan datan määrä kasvaa niiden ohella. 1970-luvulla kehitetty tietokantojen relaatiomalli ei skaalaudu hyvin yhä kasvaviin datamääriin (Jatana ym. 2012; Pokorný 2013). Jatana ym. (2012) ja Yafoozin ym. (2013) mukaan suuri osa datasta on rakenteetonta ja rakenteettoman datan hallinta tuottaa vaikeuksia relaatiomallisille tietokantajärjestelmille, koska relaatiomallinen data on luonnostaan rakenteellista ja rakenteettomaan dataan kohdistuvat SQL-kyselyt voivat olla hitaita ja monimutkaisia. Finto-palvelu (2023) määrittelee rakenteettoman datan sellaiseksi dataksi, jolla ei ole metatietojen avulla määritettyä rakennetta, jonka mukaan sen voisi jäsentää. Tällaista dataa ovat esimerkiksi kuvat, videot, äänitiedostot, verkkosivut ja tekstitiedostot. 2000-luvun aikana NoSQL-tietokannat ovat nousseet varteenotettavaksi vaihtoehdoksi perinteiselle relaatiomallille paremman skaalautuvuutensa ja rakenteettomaan dataan soveltuvuutensa takia (Mohamed, Altrafi ja Ismail 2014).

Videopelit ovat tietokoneohjelmistoja, mutta videopelinkehitys nähdään monesti erillisenä perinteisestä ohjelmistokehityksestä. Fachada (2018) kertoo videopelinkehityksen opiskelijoilla olevan huonosti motivaatiota tai kiinnostusta tietokantaopintoihin. Fachada (2018) näkee motivaation puutteen johtuvan tietokantojen näennäisestä etäisyydestä videopelien konkreettisiin ruudulla näkyviin muutoksiin. O’Grady (2021) mukaan videopeleissä ei välttämättä käytetä tietokantoja ollenkaan tai ainakaan niitä ei hyödynnetä täydessä potentiaalisuudessaan. Yhdeksi syyksi videopelien ja tietokantojen etäisyydelle O’Grady (2021) esittää olio-relaatioyhteensopimattomuuden (engl. *impedance mismatch*) relaatiomallin ja videopelinkehityksessä yleisesti käytetyn olio-ohjelmoinnin välillä.

Paikallisessa ympäristössä pelattavissa videopeleissä tietokantajärjestelmän käyttö ei ole välttämätöntä, mutta massiivisten monen pelaajan verkkopelien (engl. *massively multiplayer online game*) eli MMO-pelien jaettu ja pysyvä (engl. *persistent*) pelimaailma vaatii keskitettyä tiedonhallintaa. Kuten muussakin ohjelmistokehityksessä, Diao (2017) toteaa MMO-pelien tietokantojen olevan yleisimmin relaatiomallisia, mutta NoSQL-järjestelmien yleistymisen

on näkynyt myös MMO-pelinkehityksessä ja sitä tutkivassa akateemisessa kirjallisuudessa. Diao (2017) uskoo pilvipohjaisen arkkitehtuurin ja NoSQL-tietokantajärjestelmän hyödyntämisen olevan ratkaisevassa asemassa MMO-pelinkehityksessä skaalautuvuuden ja suorituskyvyn parantamiseksi.

Tässä kirjallisuuskatsauksessa pyritään selvittämään, miten NoSQL-tietokantoja voidaan hyödyntää MMO-pelinkehityksessä, ja olisivatko ne perinteisiä relaatiotietokantoja parempi vaihtoehto MMO-arkkitehtuurissa. Luvussa 2 annetaan taustatietoa relaatio- ja NoSQL-tietokannoista sekä niihin liittyvistä oikeellisuismalleista. Luvussa 3 avataan MMO-pelin määrittelmää pelimaailman pysyvyyden kautta ja tutustutaan lyhyesti MMO-pelien palvelinarkkitehtuuriin. Luvussa 4 tutkitaan NoSQL-tietokantojen hyödyntämistä MMO-arkkitehtuurissa. Luvussa 5 on yhteenveto ja pohdintaa MMO-tietokantojen tulevaisuudesta.

2 Tietokannat

Mikä tahansa järjestetty kokoelma yhtenäistä dataa on tietokanta, mutta tavallisesti tietokannalla tarkoitetaan jonkin loogisen tietomallin mukaiseen rakenteeseen tiettyä tarkoitusta varten tallennettua dataa, jota hallitaan tietokannan hallintajärjestelmällä (Taipalus 2021). Tässä tutkielmassa tietokannan hallintajärjestelmä lyhennetään tietokantajärjestelmäksi, vaikka esimerkiksi Taipalus (2021) viittaa tietokantajärjestelmällä tietokannan, siihen liittyvien sovellusten ja tietokannan hallintajärjestelmän muodostamaan kokonaisuuteen. Tietokannoille on monia erilaisia loogisia malleja, mutta relaatiomallin vuosikymmeniä kestäneen valta-aseman johdosta tietokannat useimmiten luokitellaan relaatiotietokantoihin ja relaatiomallista poikkeaviin NoSQL-tietokantoihin. Tässä luvussa ei perehdytä erilaisiin NoSQL-tietokantamalleihin, vaan keskitytään relaatio- ja NoSQL-tietokantojen ominaisuuksiin ja eroihin yleisemmällä tasolla.

2.1 Relaatiotietokannat

Relaatiotietokannat perustuvat Edgar Coddin (1970) esittämään loogisen tason relaatiomalliin, jossa tieto tallennetaan riveistä ja sarakkeista koostuviin tauluihin, jotka yhdistyvät toisiinsa viiteavainten avulla. Chamberlin ja Boyce (1974) kehittivät SEQUEL-kielen, josta muotoutui standardoitu SQL-kieli (*Structured Query Language*), jota käytetään datan hallinnoimiseen ja hakemiseen relaatiotietokannasta. Relaatiotietokannoilla on hyvin määritelty skeema eli ennalta suunniteltu rakenne tallennettavalle datalle (Mohamed, Altrafi ja Ismail 2014).

Relaatiotietokannan transaktiot, pienet kokoelmat käyttäjän tietokantaan tekemiä hakuja tai muutoksia, tapahtuvat ACID-oikeellisuusmallin mukaisesti (Haerder ja Reuter 1983). ACID tulee sanoista atomisuus (engl. *Atomicity*), eheys (engl. *Consistency*), eristyneisyys (engl. *Isolation*) ja pysyvyys (engl. *Durability*). On tärkeä huomata, että tässä tutkielmassa myös MMO-pelimaailmoihin liittyvä englanninkielinen *persistence*-käsite on suomennettu pysyvyudeksi. Haerder ja Reuter (1983) määrittelevät ACID-periaatteet seuraavasti: Atomisuuden mukaan transaktio suoritetaan kokonaan tai ei ollenkaan. Eheys takaa tietokannan säily-

vän tietokannan sisäisiä sääntöjä ja rajoitteita noudattavassa tilassa transaktion suorituksessa. Eristyneisyys tarkoittaa, että transaktiot tapahtuvat riippumattomina muista samanaikaisista transaktioista. Pysyvyys takaa, että suoritettujen transaktion seuraukset säilyvät tietokannassa jopa vikatilanteessa, jolloin transaktion seurauksia voi muuttaa vain toisella transaktiolla.

McKinsey & Company -konsulttiyhtiön julkaisemassa raportissa todettiin yrityksissä ja järjestöissä käsiteltävien datamäärien kasvavan räjähdysmäisesti (Manyika ym. 2011). Jatanan ym. (2012) sekä Pokornýn (2013) mukaan relaatiotietokannat skaalautuvat huonosti näihin valtaviin datamääriin. Relaatiotietokantaa voidaan skaalata lähinnä vertikaalisesti, eli lisäämällä palvelimeen resursseja, kuten muistia ja prosessoreita (Pokorný 2013). Pokorný (2013) kirjoittaa ACID-ominaisuuksien myös saattavan heikentää suorituskykyä, koska jokainen transaktio vaatii enemmän prosessointia oikeellisuuden varmistamiseksi. Datan määrän lisäksi Yafooz ym. (2013) korostavat suurimman osan käsiteltävästä datasta olevan rakenteetonta dataa, kuten pdf-tiedostoja, kuvia ja videoita. Jatanan ym. (2012) ja Yafooz ym. (2013) väittävät rakenteettoman datan soveltuvan huonosti relaatiomalliseen tietokantaan, koska relaatiomallinen data on luonnostaan rakenteellista ja rakenteettomaan dataan kohdistuvat SQL-kyselyt voivat olla hitaita ja monimutkaisia.

2.2 CAP-teoreema ja BASE-oikeellisuusmalli

Fox ja Brewer (1999) esittivät otaksunan CAP-teoreemasta, jonka Gilbert ja Lynch (2002) todistivat formaalisti. CAP-teoreeman mukaan hajautettu verkkopalvelu voi toteuttaa vain kaksi kolmesta tyypillisesti toivotusta ominaisuudesta: eheys (engl. *consistency*), saatavuus (engl. *availability*) ja pirstoutumisen sietokyky (engl. *partition tolerance*). CAP-teoreemassa eheys tarkoittaa sitä, että kaikissa verkon solmuissa (engl. *node*) on aina saatavilla yhtenäinen uusin versio datasta (Brewer 2012). Huomattavaa on, että tämä eheyden käsite eroaa luvussa 2.1 esitellystä samannimisestä ACID-ominaisuudesta. Tässä tutkielmassa eheydellä viitataan ensisijaisesti CAP-ehyteen. Gilbert ja Lynch (2002) kuvaavat saatavuuden järjestelmän kykyä vastata jokaiseen sille lähetettyyn pyyntöön. Gilbertin ja Lynchin (2002) sekä Pokornýn (2013) mukaan pirstoutumista sietävä järjestelmä toimii normaalisti, vaikka jotkin järjestelmän osat olisivat saavuttamattomissa.

Tieteellisessä kirjallisuudessa BASE-oikeellisuusmallin keksijän väitetään usein olevan Pritchett (2008) tai jopa Vogels (2008), joista jälkimmäinen ei edes käyttänyt artikkelissaan BASE-termiä, vaan kirjoitti lopulta eheästä (engl. *eventual consistency*) mallista. Todellisuudessa BASE-mallin esittivät ensimmäisenä Fox ym. (1997) vaihtoehtona ja vastakohtana ACID-mallille. BASE-lyhenne tulee seuraavista käsitteistä: yleisesti ottaen saatavissa (engl. *Basically Available*), ei aina eheä (engl. *Soft state*) ja lopulta eheä (engl. *Eventually consistent*). BASE-malli siis priorisoi CAP-teoreeman mukaisesti palvelun saatavuuden ACID-mallin suosiman eheyden kustannuksella (Fox ym. 1997). Pritchett (2008) kuvaa BASE-mallia optimistiseksi, koska se hyväksyy tietokannan eheyden olevan muuttuvassa tilassa ja eheään tilaan palataan lopulta.

2.3 NoSQL-tietokannat

Strozzi (1998) viittasi NoSQL-termillä alunperin relaatiotietokantajärjestelmään, joka ei käytä SQL-kieltä. Nykyään NoSQL-tietokannalla tarkoitetaan tietokantaa, joka ei noudata relaatiomallia. NoSQL-tietokannat eivät tyypillisesti takaa ACID-ominaisuuksia, mikä mahdollistaa merkittävästi relaatiotietokantoja paremman horisontaalisen skaalautuvuuden ja suorituskyvyn (Jatana ym. 2012; Pokorný 2013). Horisontaalinen skaalaaminen tarkoittaa järjestelmän laskentatehon tai kapasiteetin parantamista lisäämällä uusia laitteita sen sijaan, että olemassaolevia laitteita päivitetäisiin. Horisontaalisesti skaalautuvissa järjestelmissä voidaan luonnollisesti käyttää merkittävästi edullisempaa laitteistoa kuin vertikaalisesti skaalautuvissa ja skaalaamista voidaan jatkaa lähes rajattomasti (Pokorný 2013; MongoDB 2023). NoSQL-tietokannoilla ei yleensä ole ennalta määrättyä skeemaa (Jatana ym. 2012; Matallah, Belalem ja Bouamrane 2021), mikä Jatanan ym. (2012) mukaan tekee NoSQL-tietokannoista relaatiotietokantoja joustavampia. Lisäksi Jatana ym. (2012), Pokorný (2013) ja Mohamed, Altrafi ja Ismail (2014) esittävät NoSQL-tietokantojen soveltuvan relaatiotietokantoja paremmin rakenteettomalle datalle.

NoSQL-tietokantojen nähdään yleisesti toteuttavan ACID-ominaisuuksien sijaan BASE-ominaisuudet, kuten myös Jatana ym. (2012) ilmaisevat asian. Mohamed, Altrafi ja Ismail (2014) esittävät NoSQL-tietokantojen edustavan kirjoa ACID- ja BASE-mallien välillä, mikä on ehkä osuvampi kuvaus, sillä Pokornýn (2013) mukaan NoSQL-tietokannat eivät välttämät-

tä luovu kokonaan ACID-ominaisuuksista. NoSQL-tietokannat edustavat kuitenkin BASE-filosofiaa siinä mielessä, että ne painottavat saatavuutta eheyden kustannuksella (Pokorný 2013; Jatana ym. 2012; Matallah, Belalem ja Bouamrane 2021). Pokorný (2013) painottaa, että vaikka ACID-transaktiot ovat tarpeellisia kohdealueissa, joissa vaaditaan ehdotonta datan oikeellisuutta, kuten pankit ja osakemarkkinat, suorituskyky ja skaalautuvuus ovat monissa muissa tapauksissa kysytympiä. Pokornýn (2013) mukaan nämä ominaisuudet ja soveltuvuus rakenteettoman datan hallintaan tekevät NoSQL-tietokannoista erityisen hyödyllisiä pilvipalveluissa. Tietokantajärjestelmän valinta on aina tärkeää tehdä kohdealueen tarpeiden mukaan. Huomattavaa on myös NoSQL-tietokantojen suhteellisen lyhyt historia reaaliaikaisiin tietokantoihin verrattuna, minkä seurauksena kehittäjillä saattaa olla vähemmän osaamista NoSQL-teknoologioista. NoSQL-tietokantajärjestelmillä ei myöskään ole SQL-kielen kaltaista yhteistä standardoitua kyselykieltä.

3 MMO-pelien anatomia

MMO-videopeleissä samalla palvelimella voi pelata yhtäaikaisesti suuri määrä pelaajia jaetuissa pelimaailmassa. Ei ole olemassa tarkkaa määritelmää sille, mikä erottaa MMO-pelit muista monen pelaajan verkkopeleistä. MMO-peleille tyypillisiä piirteitä ovat valtava avoin pelimaailma, tavallista suurempi lukumäärä pelaajia vuorovaikuttamassa suoraan toistensa kanssa sekä edistymisen säilyminen pelisessioiden välillä, mutta näitä piirteitä esiintyy myös muissa kuin MMO-peleissä. Ehkä olennaisin MMO-pelejä määrittävä tekijä on pelimaailman pysyvyys (engl. *persistence*), jonka mukaan esimerkiksi White ym. (2007) jaottelevat verkkopelit pysyviin ja ei-pysyviin peleihin.

3.1 Pysyvät ja ei-pysyvät pelit

Whiten ym. (2007) mukaan ei-pysyvissä verkkopeleissä pelitila on sessiokohtainen, joten pelisession loppuessa tai pelaajan lähtiessä pelitilaa ei säilytetä. Pelitilalla tarkoitetaan pelimaailmassa tietyllä ajanhetkellä vallitsevia olosuhteita, kuten pelihahmojen ja esineiden sijaintia ja muita ominaisuuksia. Poikkeuksena White ym. (2007) mainitsevat joidenkin ei-pysyvien pelien, kuten Diablo 2:n, säilyttävän pelisessioiden välillä tietoa esimerkiksi pelaajan hahmon taidoista ja varusteista, mutta tällaisetkaan pelit eivät säilytä pelitilaa kokonaisuudessaan. Ei-pysyvät pelimaailmat ovat olemassa vain pelaajien vuorovaikuttaessa niiden kanssa.

Pysyvän pelimaailman pelitila on jatkuvasti tallennettuna ja ylläpidettynä riippumatta siitä, onko kukaan pelaamassa peliä vai ei. Girvanin (2018) pysyvää virtuaalista maailmaa kuvaavaa esimerkkiä mukaillen: pelaajan A pelimaailmaan jättämän esineen voi löytää toinen pelaaja B riippumatta siitä, onko pelaaja A enää maailmassa vai ei. Jos pelaaja B poimii esineen itselleen, ei pelaaja A enää löydä sitä palatessaan pelimaailmaan. Tämä toiminnallisuus saavutetaan MMO-peleissä useimmiten käyttämällä asiakas-palvelin -arkkitehtuuria (White ym. 2007; Girvan 2018; Kavalionak ym. 2015).

3.2 Asiakas-palvelin -arkkitehtuuri

Yksinkertaistettuna asiakas-palvelin -järjestelmässä asiakaslaitteet pyytävät palvelimilta resursseja jonkin tietoliikenneprotokollan mukaisilla viesteillä ja palvelimet vastaavat toimitamalla pyydetty resurssit asiakkaille (Koskimies 2005). Edellä mainitussa esimerkissä palvelin pitäisi tallessa senhetkistä pelitilaa esimerkiksi palvelimen keskusmuistissa toimivassa tietokannassa huolehtien, ettei esine näy enää maassa muille pelaajille sen jälkeen, kun pelaaja B on ottanut sen. Koska pelitilan muutoksen täytyy tapahtua kaikille pelaajille yhtäaikaaisesti, palvelin ei voi vastata asiakkaiden pyyntöihin välittömästi. Sen sijaan palvelin päivittää pelitilan kaikille pelaajille tasaisin väliajoin. Näistä päivityksistä ja niiden taajuudesta käytetään usein englanninkielisiä nimityksiä *tick* ja *tick rate* (White ym. 2007; Lee ja Chang 2015). Nopeatempoisten ensimmäisen persoonan ammuntopelien pelitilaa päivitetään sekunnin murto-osan välein. Overwatch 2 -pelissä painamalla näppäinyhdistelmää Ctrl+Shift+N peli näyttää päivitystaajuuden PPS- eli pakettia per sekunti -arvona, joka on pelin toimiessa normaalisti 64 hertsiä. Randall (2020) kertoo Valorantin palvelinten päivittävän pelitilan jopa 128 kertaa sekunnissa. Caon ym. (2011) mukaan MMO-palvelimet suorittavat noin kymmenen päivitystä sekunnissa, mutta päivitystaajuudet voivat olla vielä hitaampiakin. Esimerkiksi Runescapessa päivitykset suoritetaan 0,6 sekunnin välein ja EVE Onlinessa päivitysten välillä kuluu täysi sekunti.

MMO-palvelimilta vaaditaan skaalautuvuutta yhä suurempiin pelaajamääriin. Kavalionak ym. (2015) toteavat tämän olevan haaste asiakas-palvelin -mallin keskitetyille arkkitehtuurille. Kavalionakin ym. (2015) mukaan palvelinten kuormitusta voidaan tasoittaa jakamalla pelimaailma erillisiin alueisiin, toisintamalla pelimaailma useilla palvelimilla tai yhdistämällä molempia tekniikoita. Toisinnut pelimaailmat tarjoavat lisäksi pelaajille mahdollisuuden valita itseään maantieteellisesti läheisempi palvelin, mikä lyhentää merkittävästi pelaajan laitteen ja palvelimen välistä viivettä. Kuten Kavalionak ym. (2015) huomauttaa, nämä keinot eivät vastaa pelaajamäärän vaihtelun aiheuttamiin ongelmiin. Suuri pelaajamäärä vaatii paljon palvelimia, mutta pelaajien määrän vähentyessä merkittävästi osa palvelimista saattaa olla täysin käyttämättöminä. Yksi ratkaisu on vuokrata resursseja tarpeen mukaan pilvipalveluntarjoajalta (Buyya ym. 2009). Asiakas-palvelin -arkkitehtuurin heikkouksia on mahdollista paikata myös hyödyntämällä vertaisverkkoa, jossa käyttäjien omat laitteet toimivat

palvelimina muille käyttäjille. Tällöin tarjolla olevien resurssien määrä kasvaa luonnollisesti pelaajamäärän mukana. Esimerkiksi Barri, Roig ja Giné (2016) esittelevät asiakas-palvelin-arkkitehtuuria ja vertaisverkkoa yhdistävän hybridimallisen arkkitehtuurin MMO-peleille tavoitteenaan parempi skaalautuvuus isoihin pelaajamäärän vaihteluihin.

4 NoSQL-tietokannat MMO-pelinkehityksessä

Diao (2017) mukaan suositut MMO-pelit, kuten Second Life, World of Warcraft ja Guild Wars ovat tyypillisesti käyttäneet relaatiotietokantajärjestelmiä, joilla on ollut vaikeuksia vastata yhä kasvaviin datamääriin. Relaatiotietokantojen rajoitteista Diao (2017) nostaa esiin vaikeudet skaalaamisessa ja saatavuuden takaamisessa, tietokantojen skeemojen monimutkaisuuden ja joustamattomuuden sekä kalliista lisensseistä ja laitteistosta johtuvat korkeat kustannukset. Kuten muussakin ohjelmistokehityksessä, näihin ongelmiin on yritetty hakea ratkaisuja NoSQL-tietokannoista. Kasenides ja Paspallis (2019) näyttävät kartoituksensa perusteella NoSQL-tietokantojen käytön tutkimuksissa MMO-arkkitehtuurista olleen selkeä trendi vuodesta 2010 eteenpäin. Diao (2017) kuitenkin muistuttaa, että NoSQL-tietokantojen vahvuudet suorituskyvyn, saatavuuden, skaalautuvuuden, pirstoutumisen sietokyvyn ja kustannusten suhteen tulevat luotettavuuden ja datan eheyden kustannuksella. Lisäksi Diao (2017) mainitsee NoSQL-tietokantojen myös olevan relaatiotietokantoja uudempaa ja vähemmän kehittyntä teknologiaa, ja että NoSQL-tietokannat ovat rajallisempia osan toiminnallisuudesta, kuten kyselyjen, data-analyysin ja transaktioiden hallinnan suhteen.

4.1 Pilvipalvelut ja mikropalveluarkkitehtuuri

MMO-pelien yhteydessä NoSQL-tietokantajärjestelmistä puhuttaessa keskitytään usein kaupallisiin pilvipalveluihin. Kasenides (2023) toteaa, että kaupallisten pilvipalveluiden potentiaalia MMO-pelinkehityksessä on selvitetty suhteellisen vähän. Kasenides (2023) myös huomauttaa, että NoSQL-pilvipalveluiden tarjoajat asettavat palveluilleen rajoituksia dataobjektien koolle, mikä puolestaan rajoittaa myös tietokantaan tallennettavan MMO-pelimaailman kokoa. Kaneidesin (2023) mukaan tällaisia rajoituksia voidaan kuitenkin kiertää esimerkiksi tallentamalla pelimaailma erillisinä palasina (engl. *chunk*). Kokonaisuuksien osittaminen on MMO-pelinkehitykselle tyypillistä myös mikropalveluarkkitehtuurin muodossa (Kasenides 2023; Schweigert ym. 2023).

Kalske, Mäkitalo ja Mikkonen (2018) määrittelevät mikropalvelut arkkitehtuuriksi, jossa palvelu jaetaan pieniksi itsenäisiksi palveluiksi, joista jokainen keskittyy hoitamaan yksit-

täistä toiminnallisuutta. Kalskeen, Mäkitalon ja Mikkosen (2018) mukaan mikropalveluarkkitehtuuri kykenee hyödyntämään pilvipalveluiden automaattista skaalautuvuutta tehokkaasti. NoSQL-tietokannat saattavat olla relaatiotietokantoja sopivampia mikropalveluarkkitehtuuriin, sillä Kalske, Mäkitalo ja Mikkonen (2018) toteavat, että mikropalveluarkkitehtuurissa on suositeltavaa jakaa myös tietokanta useammaksi erilliseksi tietokannaksi, ja että ACID-ominaisuuksien saavuttaminen tällaisessa arkkitehtuurissa on vaikeaa. Myös Schweigert ym. (2023) painottavat tietokannan jakautumisen olevan vaikeaa toteuttaa relaatiotietokantajärjestelmissä. Lisäksi Kalske, Mäkitalo ja Mikkonen (2018) nostavat esiin eri tietokantateknologioiden käyttämisen palvelukohtaisesti, mikä mahdollistaisi NoSQL-tietokantojen hyödyntämisen relaatiotietokantojen rinnalla.

Diaon ym. (2014) CloudCraft-projektissa esitetään ja testataan pilvipohjaisia datanhallintaratkaisuja MMO-peleille. CloudCraft-projektin testeissä ja vertailuissa (Diao ym. 2014; Diao ym. 2015; Diao 2017) on havaittu pilvipohjaisen NoSQL-tietokantajärjestelmän Cassandraan olevan varteenotettava vaihtoehto perinteisille relaatiotietokannoille MMO-peleissä. Diao (2017) totesi Cassandraan saavuttavan testiympäristössä paremman suorituskyvyn ja skaalautuvuuden relaatiomalliseen MySQL-tietokantajärjestelmään verrattuna. Vastaavia tuloksia suorituskyvystä saavuttivat myös Kim, Han ja Yang (2016). Diao ym. (2015; 2017) toisaalta painottavat, etteivät NoSQL-tietokannat kykene välttämättä vastaamaan kaikkiin MMO-pelien datahallinnan vaatimuksiin, eivätkä NoSQL-tietokantojen heikkoudet datan eheyden suhteen ole helppoja ratkaista. Näistä syistä CloudCraft-projektissa ehdotetaan MMO-arkkitehtuuria, jossa hyödynnetään sekä relaatio- että NoSQL-tietokantoja eri tarkoituksiin.

4.2 Esimerkkejä NoSQL-tietokantoja hyödyntävistä peleistä

Videopelinkehitys on kaupallista ja yksityistä toimintaa eivätkä pelien kehittäjät tai julkaisijat välttämättä julkaise yksityiskohtaisia tietoja peliensä arkkitehtuurista tai kehityksessä käytetyistä teknologioista. Lisäksi MMO-pelien elinkaaret voivat olla hyvinkin pitkiä perinteisiin videopelisiin verrattuna. Suosituimmista MMO-peleistä esimerkiksi World of Warcraft julkaistiin alunperin vuonna 2004, Runescape vuonna 2001, Elder Scrolls Online vuonna 2014 ja Final Fantasy XIV vuonna 2010, joskin Final Fantasy XIV:n julkaisu epäonnistui ja se uudelleenjulkaistiin vuonna 2013 (Blizzard Entertainment 2004; Jagex Communications

Team 2021; Dyer 2013; Madsen 2023). Varsinkin MMO-pelien kaltaisten suuren mittakaavan ohjelmistojen päivittäminen käyttämään uusia teknologioita on kallista ja vaivalloista, joten nämä pelit saattavat käyttää suhteellisen vanhoja teknologioita. NoSQL-tietokantojen suosio on myös kasvanut merkittävästi vasta 2010-luvun aikana.

Epic Games -peilyhtiön julkaisemassa artikkelissa kerrotaan yhtiön kehittämässä Fortnite-pelissä hyödynnettävien NoSQL-tietokantajärjestelmiä MongoDB ja Redis (The Epic Team 2018). Fortnite-pelissä ei ole pysyvää pelimaailmaa, mutta se on erittäin suosittu monen pelaajan verkkopeli ja sillä on ollut jopa miljoonia samanaikaisia pelaajia; artikkelissa kirjoitetaan juuri saavutetusta 3,4 miljoonan samanaikaisen pelaajan ennätyksestä. Redis on keskusmuistissa toimiva tietokantajärjestelmä, jota myös Kasenides (2023) käyttää tapaus- tutkimuksensa Mars Pioneer -MMO-pelissä.

Posniewski (2008) kertoo Game Developers Conference -konferenssissa pitämässään puheessa edustamansa MMO-peleihin erikoistuneen Cryptic Studios -peilyhtiön kohtaamista haasteista relaatiotietokantojen käyttämisestä MMO-peleissä. Hän esittää ratkaisuna näihin ongelmiin Cryptic Studios -yhtiön oman Cryptic DB -tietokantajärjestelmän, jonka tavoitteina on muun muassa säilyttää ACID-ominaisuudet mahdollisimman hyvin, madaltaa viivettä, parantaa kirjoitus- ja lukunopeuksia sekä helpottaa muutoksia skeemaan ja varmuuskopiointiprosessia. Vuonna 2008 NoSQL ei vielä ollut kovin yleisesti käytetty termi, eikä Posniewski (2008) käytä tätä nimitystä Cryptic DB -tietokantajärjestelmästä, vaikkei se noudatakaan relaatiomallia tai tue SQL-kieltä.

Walsh ja Amazon Game Tech Team (2022) kertovat blogikirjoituksessa Amazonin kehittämän New World -MMO-pelin arkkitehtuurista. New World -pelin pelitilan tallentamiseen käytetään Amazonin omaa pilvipohjaista NoSQL-tietokantajärjestelmää DynamoDB. Walsh ja Amazon Game Tech Team (2022) kirjoittavat DynamoDB-tietokannan mahdollistavan helpon ja laajan skaalaamisen sekä suuren kirjoitusoperaatiomäärän käsittelyn. New World -pelin julkaisu ei kuitenkaan sujunut täysin ongelmitta. Khder ym. (2023) kertovat noin kuukausi pelin julkaisun jälkeen havaitusta virheestä, joka mahdollisti pelivaluutan ja tavaroiden kahdentamisen. Kahdentaminen toteutettiin vaihtamalla rahaa tai tavaraa kahden pelaajan välillä. Vaihtokaupan vastaanottaja hidasti tarkoituksella omien verkkopakettien lähetystä palvelimelle, ensin hyväksyi ja sitten perui vaihtokaupan ja lopulta palautti internetin toi-

minnan normaaliksi. Tästä seurasi, että vastaanottaja sai vaihdetun tavaran lähettäjän menettämättä sitä. Tällaiset kahdentamisvirheet (engl. *duplication glitch*) ovat kohtuullisen yleisiä verkkopeleissä. Khder ym. (2023) esittävät, että kahdentamisvirheet olisi mahdollista estää tallentamalla data kaikki tai ei mitään -transaktioina, eli käytännössä ACID-mallia noudattaen, vaikkeivat Khder ym. (2023) tätä termiä käytäkään.

5 Yhteenveto

Tietokannoista MMO-videopelien kontekstissa on suhteellisen vähän tieteellistä kirjallisuutta. Tietokantoja ei nähdä keskeisenä osana perinteistä pelinkehitystä ja ehkä osittain sen takia ne jäävät vähemmälle huomiolle MMO-pelinkehityksessäkin. Perinteisissä videopeleissä datan tallennukseen saatetaan varsinaisen tietokantajärjestelmän sijasta käyttää vain yksinkertaista JSON-tiedostoa. Julkaistujen videopelien käyttämät arkkitehtuurit ja teknologiat eivät myöskään usein ole helposti saatavilla, minkä takia niitä kartoittava tutkimuskin on harvinaista. Vaikka MMO-peleissä on perinteisesti käytetty vain relaatiotietokantoja, tieteellinen kirjallisuus aiheesta keskittyy lähinnä NoSQL-tietokantoihin. Tähän vaikuttaa varmasti tietotekniikan alalla samanaikaisesti kasvanut yleinen kiinnostus NoSQL-tietokantoja kohtaan.

NoSQL-tietokannoilla vaikuttaa niin tieteellisen kirjallisuuden kuin pelinkehityksen trendien pohjalta olevan potentiaalia MMO-arkkitehtuurissa. NoSQL-tietokannoilla on monia etuja relaatiotietokantoihin nähden. NoSQL-tietokannat tarjoavat relaatiotietokantoja parempaa skaalautuvuutta, saatavuutta, suorituskykyä ja joustavuutta. Nämä edut kuitenkin saavutetaan CAP-teoreeman mukaisesti heikentyvän eheyden ja ACID-ominaisuuksien takaaman luotettavuuden kustannuksella. Tietokantajärjestelmän valinta tulee aina tehdä huolellisesti kohdealueen tarpeet ja kehittäjien osaaminen huomioiden. NoSQL-tietokantojen suosio voi näkyä viiveellä MMO-pelinkehityksessä, sillä MMO-peleillä on pitkät elinkaaret, ja niiden kehittäminen ja ylläpitäminen on kallista ja työlästä. MMO-pelaajat eivät myöskään välttämättä lakkaa pelaamasta vanhaa peliä, vaikka siitä tehtäisiin uusi versio, ja saattavat jopa vaatia paluuta pelin aikaisempiin versioihin isojen muutosten jälkeen.

Ohjelmistojen tarjoaminen pilvipohjaisina palveluina on ollut kasvava trendi tietotekniikassa. MMO-peleille ominaisesta pelaajien jakamasta pysyvästä pelimaailmasta johtuen MMO-pelit ovat myös aina olleet eräänlaisia pilvipalveluita, joiden pelaamisesta on muista peleistä poiketen veloitettu jo 1990-luvun lopulta kuukausittaisia maksuja. Pilvipalveluiden trendi näkyy myös tietokannoissa, ja niin relaatio- kuin NoSQL-mallisia pilvipohjaisia tietokantajärjestelmiä on lukuisia. Pilvipalveluiden hyödyntämisestä keskustellaan paljon MMO-pelinkehityksen yhteydessä. Myös suurin osa MMO-pelien arkkitehtuuriin ja tietokantoihin liittyvästä tutkimuksesta keskittyy kaupallisiin pilvipalveluihin.

Tässä tutkielmassa käsitellyn kirjallisuuden perusteella pilvipohjainen tiedonhallinta ja NoSQL-tietokannat vaikuttavat olevan keskeisiä MMO-pelinkehityksen tulevaisuudessa. Todennäköisesti MMO-arkkitehtuurissa tullaan hyödyntämään useita eri tietokantateknologioita erilaisiin tarpeisiin, kuten käyttäjätietojen hallintaan, pelitilan ylläpitämiseen ja varmuuskopiointiin. Aiheesta on kuitenkin vain rajallisesti tieteellistä tutkimusta ja yksityiskohtaista tietoa julkaistujen pelien arkkitehtuurista on harvoin saatavilla. Tulevat MMO-pelijulkaisut näyttävät, mihin suuntaan MMO-pelinkehitys etenee ja toivon mukaan myös inspiroivat lisää tutkimusta MMO-pelien arkkitehtuurista ja tiedonhallinnasta.

Lähteet

Barri, Ignasi, Concepció Roig ja Francesc Giné. 2016. “Distributing game instances in a hybrid client-server/P2P system to support MMORPG playability”. *Multimedia Tools and Applications* 75:2005–2029.

Blizzard Entertainment. 2004. *Blizzard Entertainment Announces World of Warcraft "Street Date" - November 23, 2004*. Saatavilla WWW-muodossa, <https://web.archive.org/web/20041110022959/http://www.blizzard.com/press/040411-street-date.shtml>, viitattu 3.5.2024.

Brewer, Eric. 2012. “CAP twelve years later: How the rules have changed”. *Computer* 45 (2): 23–29. <https://doi.org/10.1109/MC.2012.37>.

Buyya, Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg ja Ivona Brandic. 2009. “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility”. *Future Generation computer systems* 25 (6): 599–616.

Cao, Tuan, Marcos Vaz Salles, Benjamin Sowell, Yao Yue, Alan Demers, Johannes Gehrke ja Walker White. 2011. “Fast checkpoint recovery algorithms for frequently consistent applications”, 265–276. SIGMOD ’11. Athens, Greece: Association for Computing Machinery. ISBN: 9781450306614. <https://doi.org/10.1145/1989323.1989352>.

Chamberlin, Donald D. ja Raymond F. Boyce. 1974. “SEQUEL: A structured English query language”. Teoksessa *Proceedings of the 1974 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control*, 249–264. SIGFIDET ’74. Association for Computing Machinery. ISBN: 9781450374156. <https://doi.org/10.1145/800296.811515>.

Codd, E. F. 1970. “A relational model of data for large shared data banks”. *Commun. ACM* 13, numero 6 (kesäkuu): 377–387. ISSN: 0001-0782. <https://doi.org/10.1145/362384.362685>.

Diao, Ziqiang. 2017. “Cloud-based support for Massively Multiplayer Online Role-Playing Games”. Tohtorinväitöskirja, Otto-von-Guericke-Universität Magdeburg. <http://dx.doi.org/10.25673/4571>.

- Diao, Ziqiang, Shuo Wang, Eike Schallehn ja Gunter Saake. 2014. "Cloudcraft: Cloud-based data management for mmorpgs". Teoksessa *Databases and Information Systems VIII*, 71–84. IOS Press.
- Diao, Ziqiang, Pengfei Zhao, Eike Schallehn ja Siba Mohammad. 2015. "Achieving Consistent Storage for Scalable MMORPG Environments", 33–40. IDEAS '15. Yokohama, Japan: Association for Computing Machinery. ISBN: 9781450334143. <https://doi.org/10.1145/2790755.2790758>.
- Dyer, Mitch. 2013. *The Elder Scrolls Online PC, Xbox One and PS4 Release Dates Announced*. Saatavilla WWW-muodossa, <https://www.ign.com/articles/2013/12/11/the-elder-scrolls-online-pc-xbox-one-and-ps4-release-dates-announced>, viitattu 3.5.2024.
- Fachada, Nuno. 2018. "Teaching database concepts to video game design and development students". *Revista Lusófona de Educação* 40. <http://hdl.handle.net/10437/9326>.
- Finto-palvelu. 2023. *Määritelmä termille "rakenteeton data"*. Saatavilla WWW-muodossa, <https://finto.fi/tt/fi/page/t53>, viitattu 9.4.2024.
- Fox, Armando ja Eric A Brewer. 1999. "Harvest, yield, and scalable tolerant systems". Teoksessa *Proceedings of the seventh workshop on hot topics in operating systems*, 174–178. IEEE.
- Fox, Armando, Steven D. Gribble, Yatin Chawathe, Eric A. Brewer ja Paul Gauthier. 1997. "Cluster-based scalable network services". *SIGOPS Oper. Syst. Rev.* 31, numero 5 (lokakuu): 78–91. ISSN: 0163-5980. <https://doi.org/10.1145/269005.266662>.
- Gilbert, Seth ja Nancy Lynch. 2002. "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services". *Acm Sigact News* 33 (2): 51–59.
- Girvan, Carina. 2018. "What is a virtual world? Definition and classification". *Educational Technology Research and Development* 66 (5): 1087–1100.
- Haerder, Theo ja Andreas Reuter. 1983. "Principles of transaction-oriented database recovery". *ACM Computing Surveys* 15, numero 4 (joulukuu): 287–317. ISSN: 0360-0300. <https://doi.org/10.1145/289.291>.

- Jagex Communications Team. 2021. *Runescape Celebrates 20 Glorious Years*. Saatavilla WWW-muodossa, <https://www.jagex.com/en-GB/news/qe0u7D/runescape-celebrates-20-glorious-years>, viitattu 3.5.2024.
- Jatana, Nishtha, Sahil Puri, Mehak Ahuja, Ishita Kathuria ja Dishant Gosain. 2012. “A survey and comparison of relational and non-relational database”. *International Journal of Engineering Research & Technology* 1 (6): 1–5.
- Kalske, Miika, Niko Mäkitalo ja Tommi Mikkonen. 2018. “Challenges when moving from monolith to microservice architecture”. Teoksessa *Current Trends in Web Engineering: ICWE 2017 International Workshops*, 32–47. Springer International Publishing. ISBN: 978-3-319-74433-9. https://doi.org/10.1007/978-3-319-74433-9_3.
- Kasenides, Nicos. 2023. “Models, methods, and tools for developing MMOG backends on commodity clouds”. Tohtorinväitöskirja, University of Central Lancashire.
- Kasenides, Nicos ja Nearchos Paspallis. 2019. “A systematic mapping study of MMOG backend architectures”. *Information* 10 (9). ISSN: 2078-2489. <https://doi.org/10.3390/info10090264>.
- Kavalionak, Hanna, Emanuele Carlini, Laura Ricci, Alberto Montresor ja Massimo Coppola. 2015. “Integrating peer-to-peer and cloud computing for massively multiuser online games”. *Peer-to-Peer Networking and Applications* 8:301–319.
- Khder, Moaiad Ahmed, Samer Shorman, Ahmed F Ali ja Mohammed O Al-Mudaifa. 2023. “Online Games Data Storing Methods, Vulnerabilities, and Solutions”. Teoksessa *2023 International Conference on IT Innovation and Knowledge Discovery (ITIKD)*, 1–5. IEEE.
- Kim, Seong-Hoon, Gi-Tae Han ja Young-Kyu Yang. 2016. “Cloud Database Design for MMORPG and Performance Analysis based on User Play Tendency”. *International Information Institute (Tokyo)* 19, numero 2 (helmikuu): 661–672.
- Koskimies, Kai. 2005. *Ohjelmistoarkkitehtuurit*. Toimittanut Tommi Mikkonen. Valikko. Talentum. <https://jyu.finna.fi/Record/keski.347017>.

- Lee, Wai-Kiu ja Rocky KC Chang. 2015. "Evaluation of lag-related configurations in first-person shooter games". Teoksessa *2015 International Workshop on Network and Systems Support for Games (NetGames)*, 1–3. IEEE.
- Madsen, Hayes. 2023. *Realm Reborn Is Still Gaming's Greatest Comeback*. Saatavilla WWW-muodossa, <https://www.inverse.com/gaming/ff14-final-fantasy-xiv-realm-reborn-10th-anniversary>, viitattu 3.5.2024.
- Manyika, James, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh ja Angela Hung Byers. 2011. "Big data: The next frontier for innovation, competition, and productivity".
- Matallah, Houcine, Ghalem Belalem ja Karim Bouamrane. 2021. "Comparative study between the MySQL relational database and the MongoDB NoSQL database". *International Journal of Software Science and Computational Intelligence (IJSSCI)* 13 (3): 38–63.
- Mohamed, Mohamed A, Obay G Altrafi ja Mohammed O Ismail. 2014. "Relational vs. nosql databases: A survey". *International Journal of Computer and Information Technology* 3 (03): 598–601.
- MongoDB. 2023. *A Guide to Horizontal vs Vertical Scaling*. Saatavilla WWW-muodossa, <https://www.mongodb.com/basics/horizontal-vs-vertical-scaling>, viitattu 5.4.2024.
- O'Grady, Daniel. 2021. "Bringing database management systems and video game engines together". Tohtorinväitöskirja, Universität Tübingen.
- Pokorný, Jaroslav. 2013. "NoSQL Databases: a step to database scalability in Web environment", 9:278–283. 1. Joulukuu. <https://doi.org/10.1108/17440081311316398>.
- Posniewski, Shannon. 2008. *SQL Considered Harmful: Databases for MMORPGs*. Saatavilla WWW-muodossa, <https://www.gdcvault.com/play/339/SQL-Considered-Harmful-Databases-for>, viitattu 10.4.2024. Game Developers Conference.
- Pritchett, Dan. 2008. "BASE: An Acid Alternative: In partitioned databases, trading some consistency for availability can lead to dramatic improvements in scalability." *Queue* 6, numero 3 (toukokuu): 48–55. ISSN: 1542-7730. <https://doi.org/10.1145/1394127.1394128>.

- Randall, Brent. 2020. *Valorant's 128-Tick Servers*. Saatavilla WWW-muodossa, <https://technology.riotgames.com/news/valorants-128-tick-servers>, viitattu 6.11.2023.
- Schweigert, Marlon H, Diego EGC de Oliveira, Guilherme P Koslovski, Maurício A Pillon ja Charles C Miers. 2023. "Experimental analysis of microservices architectures for hosting cloud-based Massive Multiplayer Online Role-Playing Game (MMORPG)". Teoksessa *2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 115–122. IEEE.
- Strozzi, Carlo. 1998. *NoSQL: a non-SQL RDBMS*. Saatavilla WWW-muodossa, http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/NoSQL/Home%20Page/, viitattu 20.3.2024.
- Taipalus, Toni. 2021. *ITKA2004 Tietokannat ja tiedonhallinta*. Saatavilla WWW-muodossa, <https://tim.jyu.fi/velp/kurssit/tktl/itka204/kurssimoniste#johdanto>, viitattu 3.5.2024.
- The Epic Team. 2018. *Postmortem of Service Outage at 3.4M CCU*. Saatavilla WWW-muodossa, <https://www.fortnite.com/news/postmortem-of-service-outage-at-3-4m-ccu>, viitattu 11.4.2024. Epic Games.
- Walsh, Nicholas ja Amazon Game Tech Team. 2022. *The Unique Architecture behind Amazon Games' Seamless MMO New World*. Saatavilla WWW-muodossa, <https://aws.amazon.com/blogs/gametech/the-unique-architecture-behind-amazon-games-seamless-mmo-new-world/>, viitattu 1.12.2023.
- White, Walker, Christoph Koch, Nitin Gupta, Johannes Gehrke ja Alan Demers. 2007. "Database Research Opportunities in Computer Games". *SIGMOD Rec.* (New York, NY, USA) 36, numero 3 (syyskuu): 7–13. ISSN: 0163-5808. <https://doi.org/10.1145/1324185.1324186>.
- Vogels, Werner. 2008. "Eventually Consistent: Building reliable distributed systems at a worldwide scale demands trade-offs? between consistency and availability." *Queue* 6 (6): 14–19.
- Yafooz, Wael MS, Siti ZZ Abidin, Nasiroh Omar ja Zanariah Idrus. 2013. "Managing unstructured data in relational databases". Teoksessa *2013 IEEE Conference on Systems, Process & Control (ICSPC)*, 198–203. IEEE.