

Optimization models for allocating memory capacity in server applications (valmiin työn esittely)

Eero Virmavirta

16.5.2023

Ohjaaja: *TkK Onni Lampi*

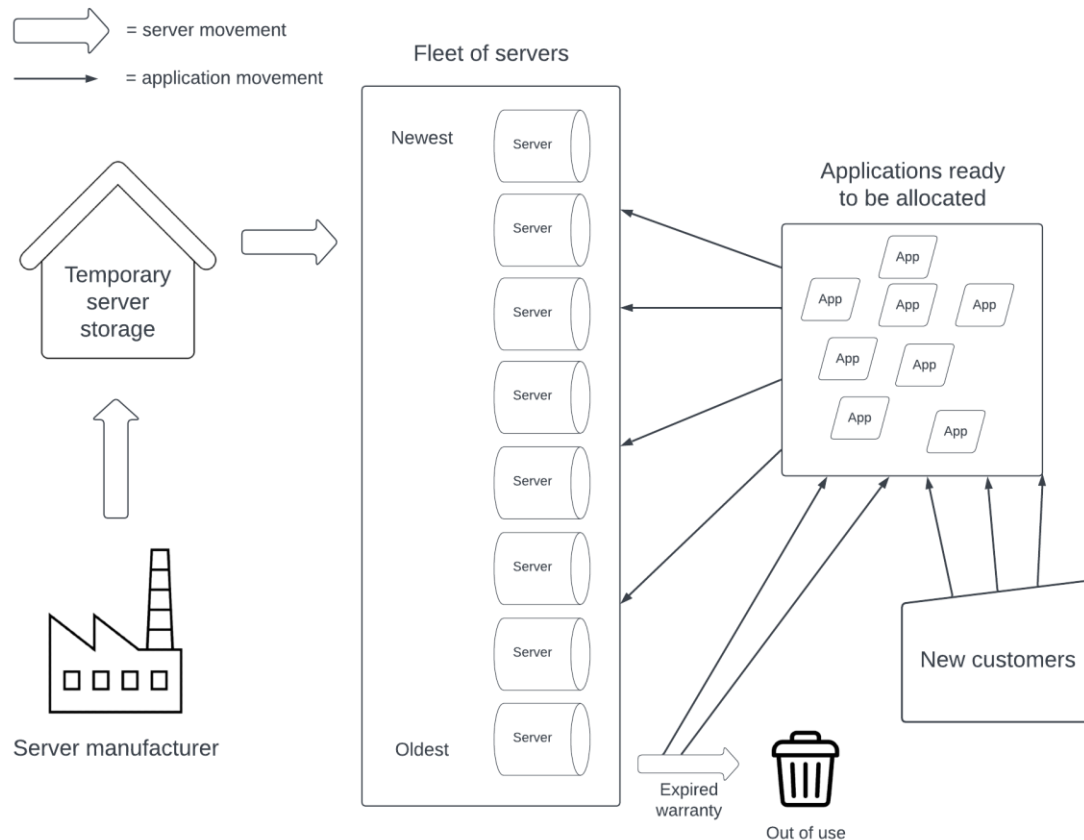
Valvoja: *Prof. Ahti Salo*

Työn saa tallentaa ja julkistaa Aalto-yliopiston avoimilla verkkosivuilla. Muilta osin kaikki oikeudet pidätetään.

Tausta: kapasiteetinhallinta

- Kapasiteetinhallinnalla pyritään varmistamaan organisaation resurssien riittävyys liiketoiminnan tarpeisiin
- Tavoitteena resurssien käytön optimointi ja resurssikustannusten minimointi
- Kapasiteetinhallintaa harjoitetaan monilla aloilla
 - Tuotanto, sairaanhoito, työvoimaoptimointi, IT-ala
- Työssä keskitytään IT-alaan ja erityisesti palvelimien muistikapasiteetin optimointiin

Tausta: allokaatioprosessi



Tavoitteet ja rajaukset

- Tarkastellaan muistin (RAM) käyttöä palvelimilla
 - Vaikeasti ennakoitava laskentateho (CPU) jätetään huomiotta
- Luodaan optimointimalli palvelimien optimaalisen muistikapasiteetin käyttöasteen selvittämiseksi
 - Sovelletaan mallia kohdeorganisaatioon
- Luodaan viisi yksinkertaista allokatiomallia palvelimien täyttämiseksi
 - Vertaillaan näitä kapasiteettitarpeen ennusteeseen

Trimmitysongelma

- Materiaali tulee jakaa ennalta määritellyn kokoisiin osiin niin, että mahdollisimman vähän materiaalia menee hukkaan
- Klassinen esimerkki: Vakiopituisista tukeista tulee sahata tietty määrä lyhyempiä, eripituisia pätkiä
- Sovellusten sijoittaminen palvelimille mallinnetaan tässä työssä trimmitysongelmana

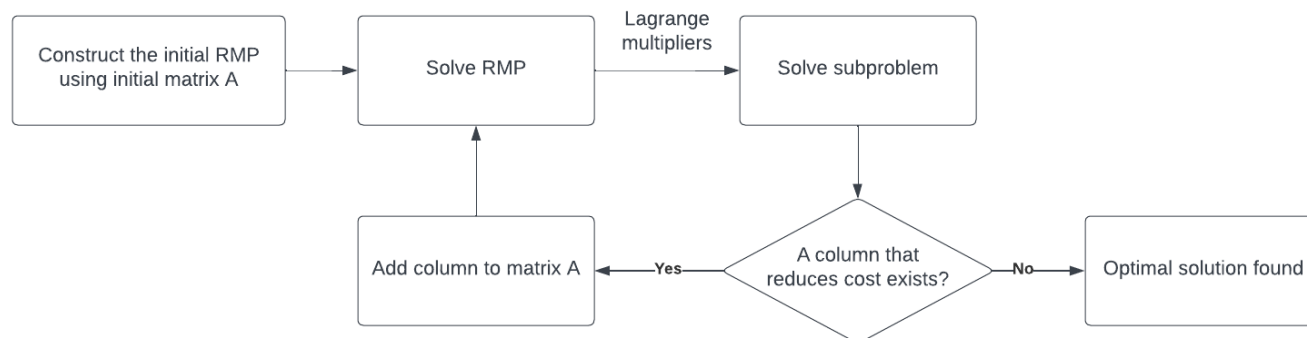
Optimointimalli

$$\begin{aligned} \min \quad & \sum_{j=1}^n c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^n a_{ij} x_j = n_i, \quad \forall i = 1, 2, \dots, m \\ & x_j \in N_0. \end{aligned}$$

- c_j = käyttämätön muisti täyttömallilla j
- x_j = kuinka monta kertaa täyttömallia j käytetään
- a_{ij} = kuinka monta s_i -kokoista sovellusta täyttömalli j käyttää
- $n_i = s_i$ -kokoisten sovellusten lukumäärä

Viivästetty sarakkeiden luominen

- Ratkaistaan ongelma osissa
- Ei sisällytetä kaikkia mahdollisia täyttömalleja, lisätään vain tulosta parantavat mallit
 - Ratkaistaan alkuperäinen ongelma rajatulla määrällä täyttömalleja (restricted master problem, RMP) → etsitään uusi täyttömalli Lagrangen kertoimia hyödyntämällä (subproblem)



Viivästetty sarakkeiden luominen

$$\begin{aligned} \min \quad & \sum_{j=1}^n x_j \\ \text{s.t.} \quad & Ax \geq n \\ & x_j \geq 0. \end{aligned}$$

Rajoitettu ongelma (RMP)

$$\begin{aligned} \max \quad & \sum_{i=1}^m \lambda_i a_i \\ \text{s.t.} \quad & \sum_{i=1}^m a_i s_i \leq S \\ & a_i \in N_0 \end{aligned}$$

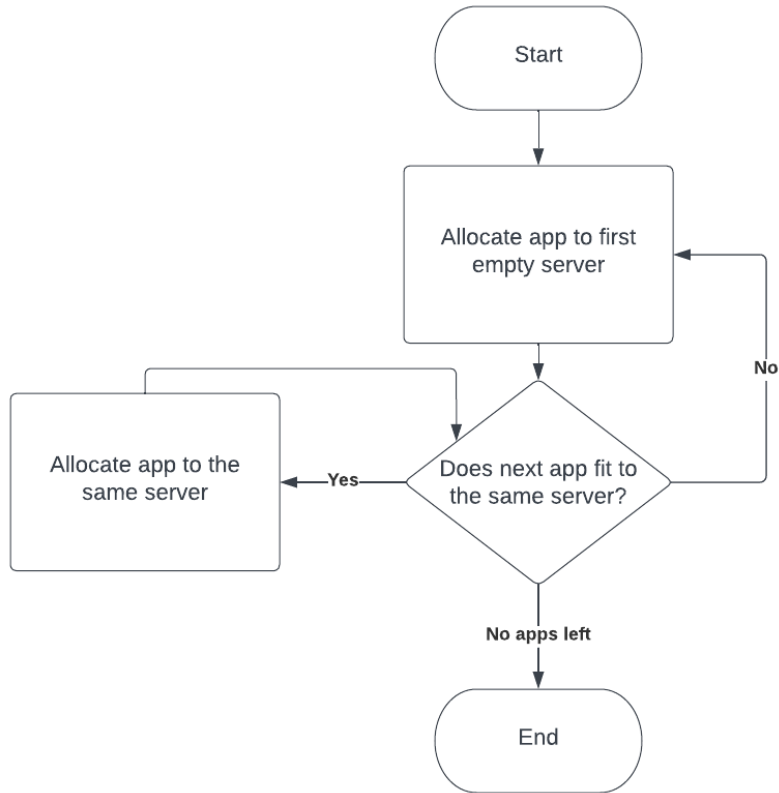
Aliongelma

- A = matriisi jonka elementteinä a_{ij}
- n = vektori jonka elementteinä n_i
- x = vektori jonka elementteinä x_j
- λ_i = RMP:n rajoitteita vastaavat Lagrangen kertoimet

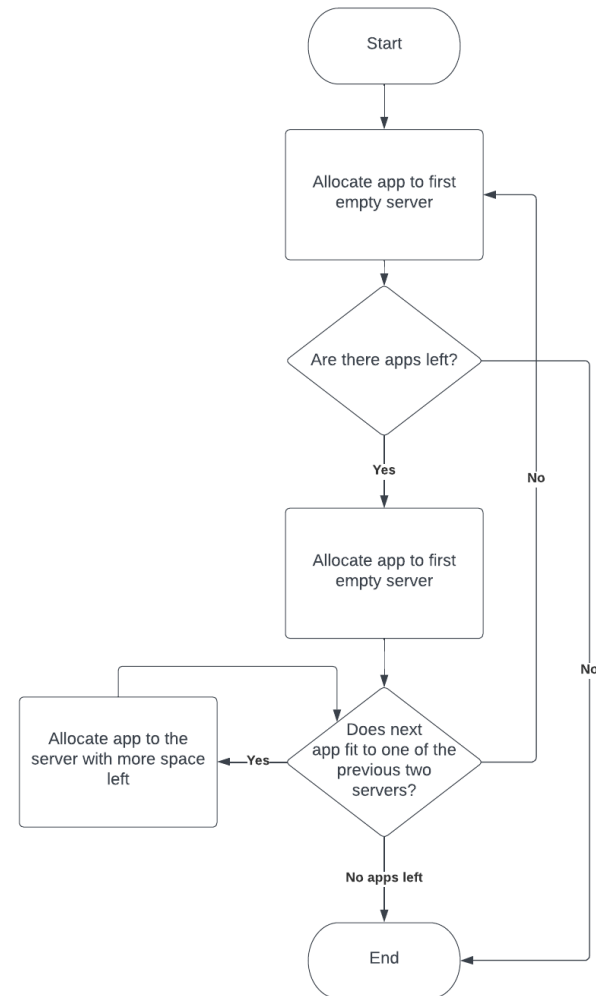
Allokaatiomallit

- Luodaan viisi yksinkertaista mallia, joilla allokoidaan sovelluksia palvelimille
- Allokaatiomalleja vertaillaan syöttämällä niille satunnaisesti generoituja sovelluksia ja tutkimalla muistikapasiteetin käyttöastetta

Allokaatiomallit

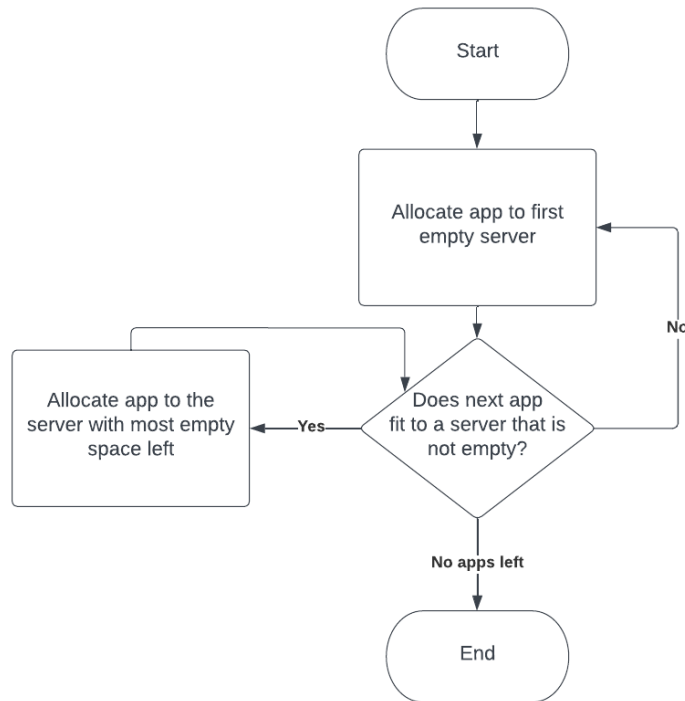


Malli A

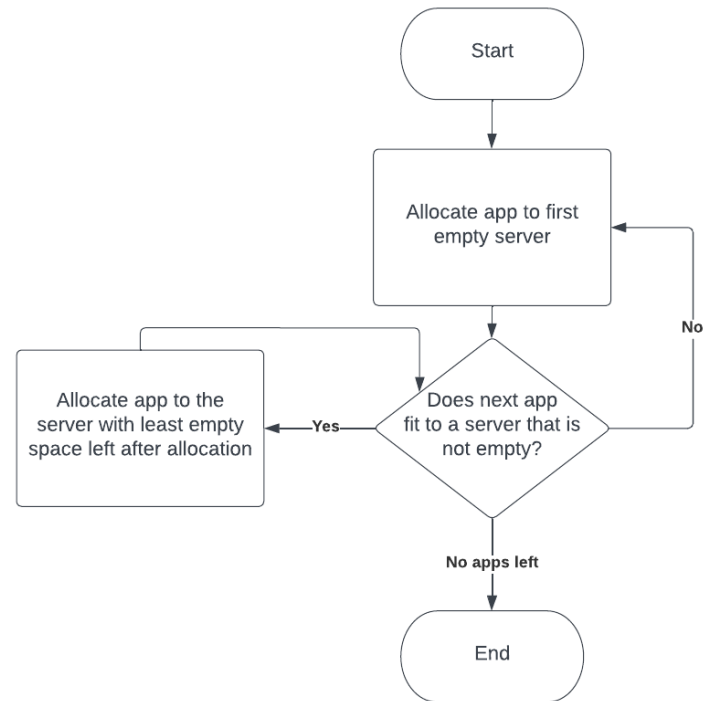


Malli B

Allokaatiomallit

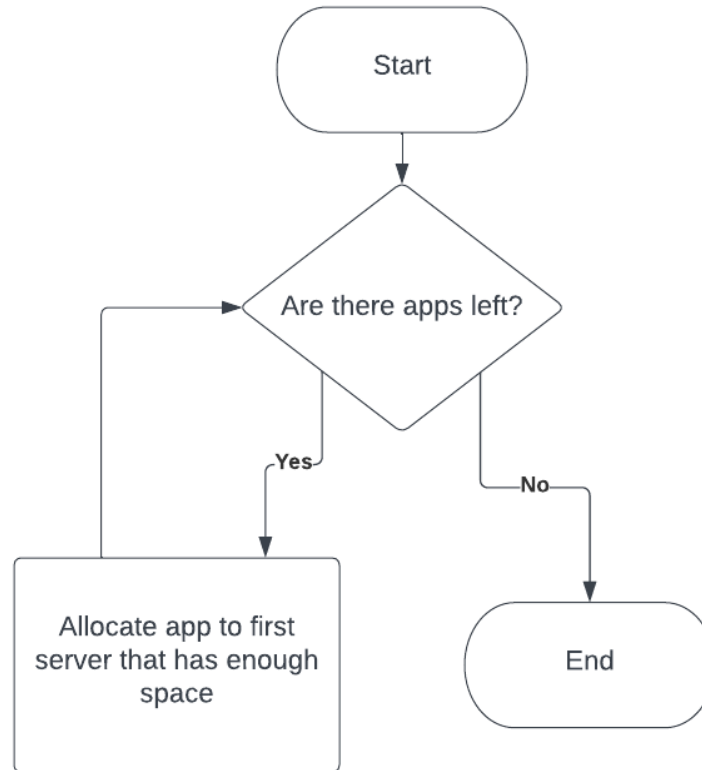


Malli C



Malli D

Allokaatiomallit



Malli E

TULOKSET

Trimmitysoptimointi

- Data perustuu kohdeorganisaation tämänhetkisiin sovelluksiin ja palvelimiin
- Eri kokoisten ympäristöjen suuri lukumäärä teki laskennasta raskaan
- Yksittäiset pienet sovellukset olivat ongelmallisia, koska sarakkeiden luominen ei huomioi sovellusten lukumääriä
→ päädyttiin osin epäoptimaalisiin täyttömalleihin

Sovellusten lukumäärä	1277
Sovellusten kokojen vaihteluväli	1 GB – 4,6 TB
Erillisiä sovellusten kokoja	193
Sovellusten kokojen keskiarvo	339 GB
Sovellusten kokojen mediaani	80 GB
Palvelimien koko	3 TB

Trimmitysoptimointi

- Yli 3 TB ja alle 300 GB sovellukset jätettiin pois lopullisesta optimoinnista
 - Pienet sovellukset lisättiin jälkikäteen ad-hoc menetelmillä

Trimmitysoptimoinnin tulokset	
Muistikapasiteetin kokonaiskäyttöaste	95,85 %
Palvelimien lukumäärä	104
Sovellusten lukumäärä	298
Sovellusten erillisten kokojen lukumäärä	78
Sovellusten koko yhteensä	299,05 TB
Tyhjää tilaa palvelimilla yhteensä	12,95 TB
Erillisten täyttömallien lukumäärä	52

Trimmitysoptimointi

- Osa pienistä sovelluksista lisättiin käsin palvelimille trimmitysoptimoinnista jääneeseen tyhjään tilaan
- Loput sovellukset allokoitiin mallilla D
- Lopullinen käyttöaste oli paras mahdollinen

Kapasiteettiallokoinnin lopulliset tulokset	
Muistikapasiteetin kokonaiskäyttöaste	99,29 %
Palvelimien lukumäärä	126
Sovellusten lukumäärä	1261
Sovellusten erillisten kokojen lukumäärä	181
Sovellusten koko yhteensä	375,33 TB
Tyhjää tilaa palvelimilla yhteensä	2,67 TB

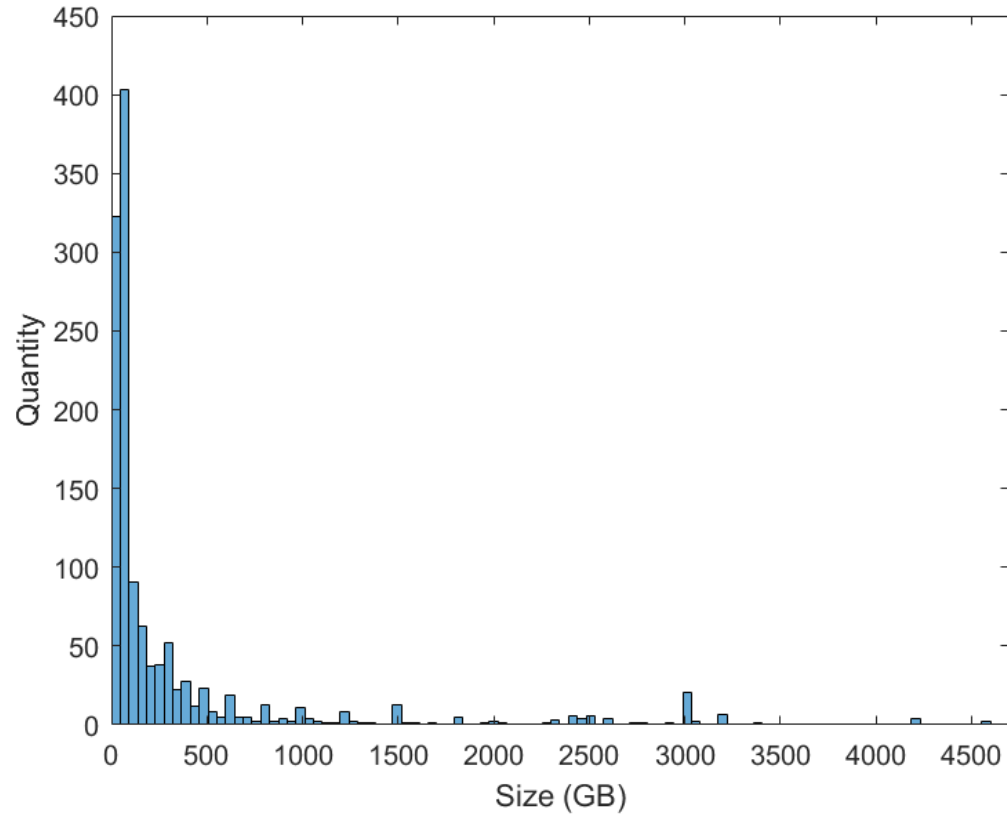
Trimmitysoptimointi: yhteenveto

- Ilman pieniä sovelluksia optimoinnilla saavutettiin korkea käyttöaste
 - Laskenta oli tehokasta
- Sarakkeiden luominen ei täysin toimiva, kun tietyn kokoisten sovellusten yhteenlaskettu koko pienempi kuin palvelimen kapasiteetti
 - Ilman sarakkeiden luomista ongelma liian suuri ratkaistavaksi

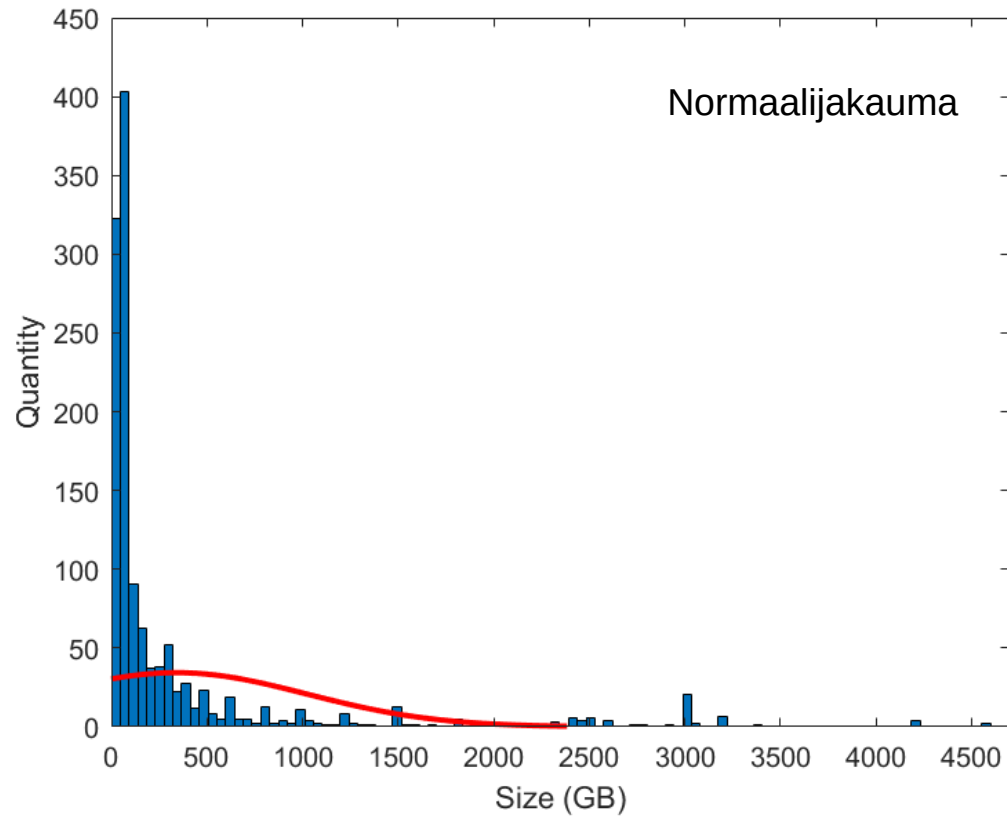
Allokaatiomallien vertailu

- Sovelluksia allokoitiin palvelimille yksi kerrallaan, yksi sovellus viikossa
- Tulevien sovellusten koko ei etukäteen tiedossa
- Sovellusten kokojen todennäköisyydet sovitettiin tämänhetkisten sovellusten jakaumaan
- Malleja vertailtiin käyttöasteen kehittymisen perusteella

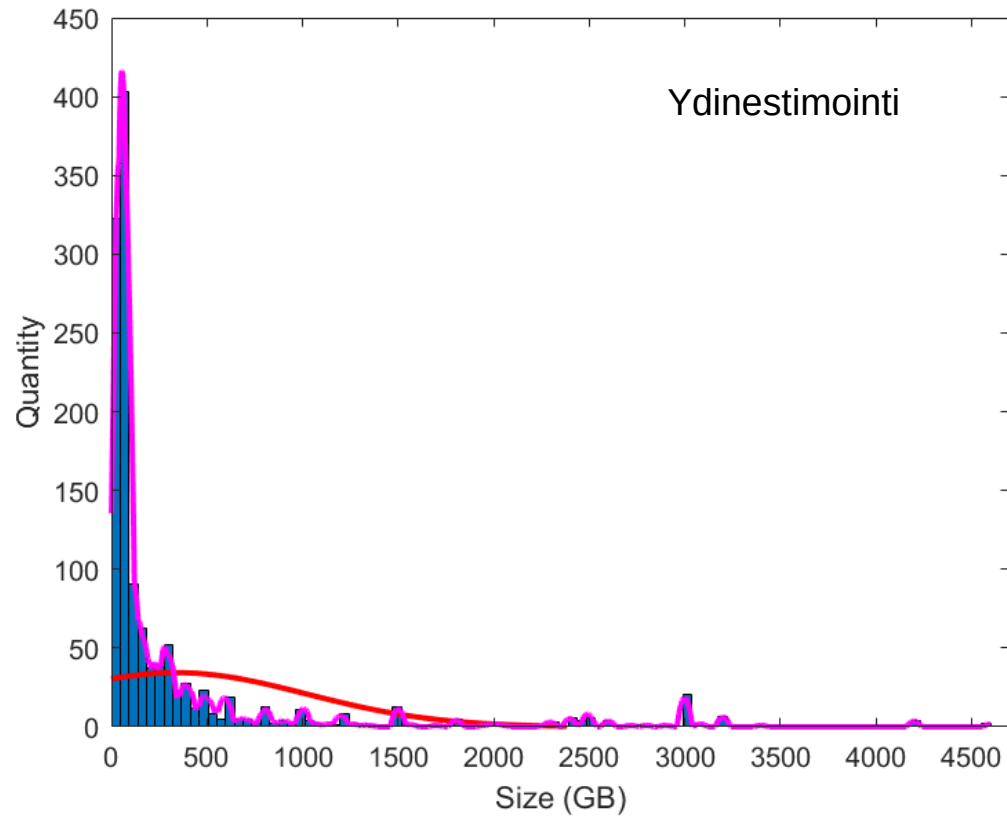
Sovellusten muistivaatimukset



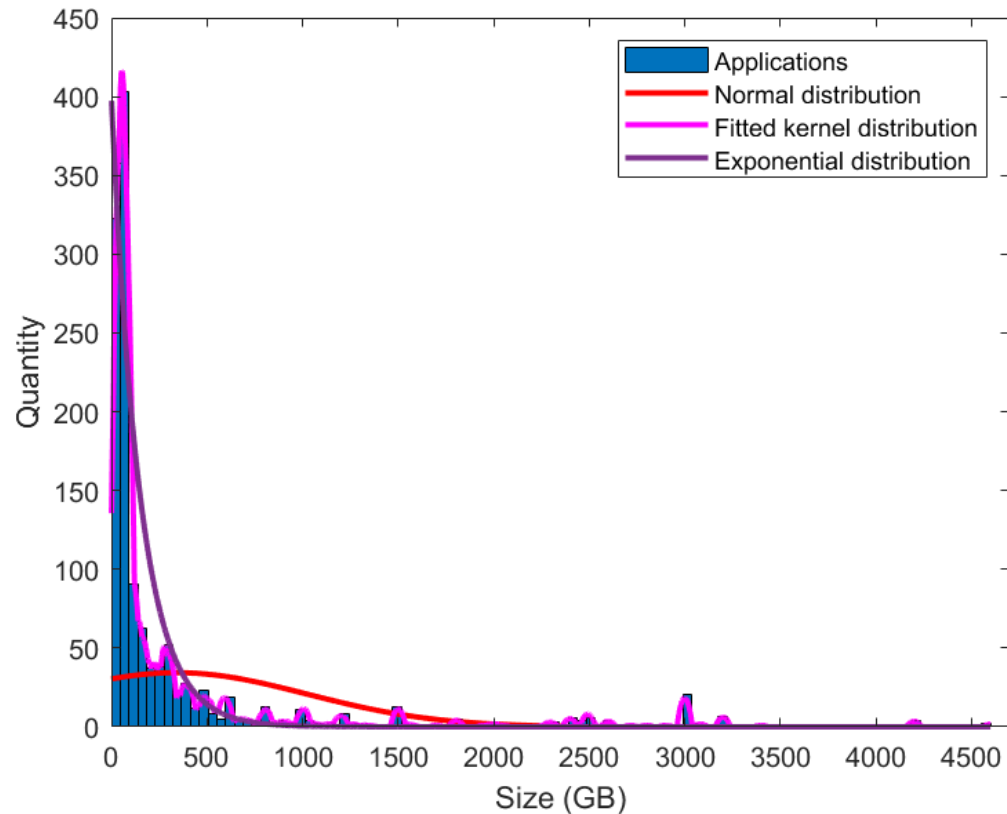
Sovellusten muistivaatimukset



Sovellusten muistivaatimukset



Sovellusten muistivaatimukset



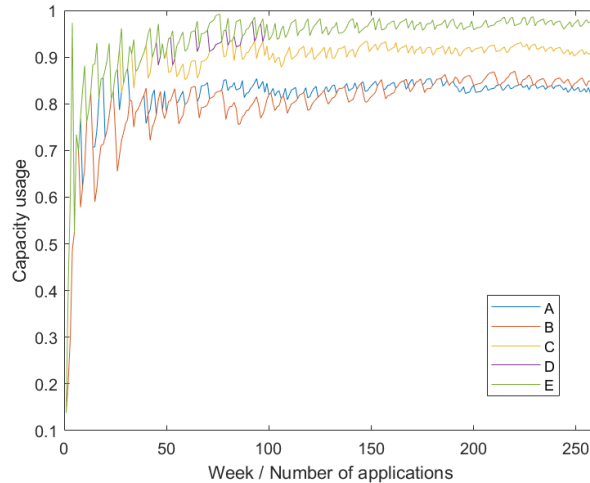
Allokaatiomallien vertailu

Käyttöasteet (%) eri jakaumilla ja allokatiomalleilla

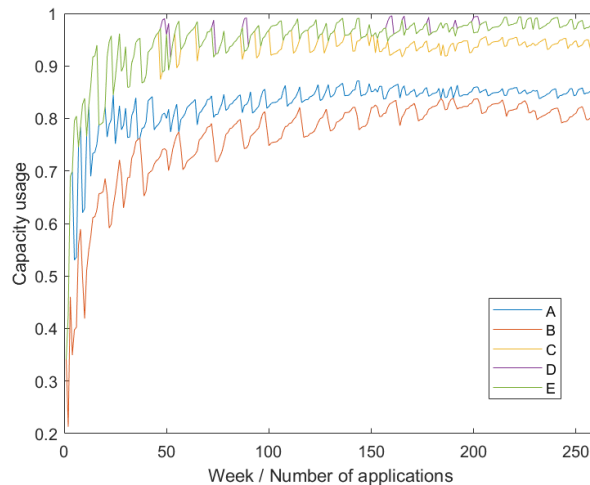
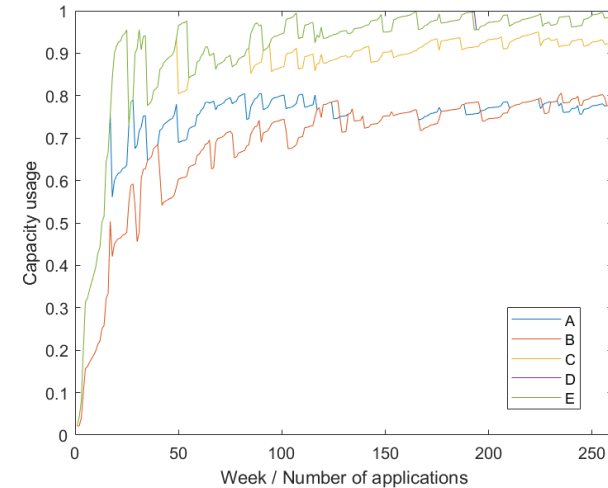
Distribution	Model	6 months	1 year	2 years	5 years	10 years
Normal	A	80,81	83,10	84,20	84,62	84,95
	B	75,97	78,82	80,87	82,34	83,17
	C	85,52	88,65	90,61	91,80	92,32
	D	89,30	93,43	96,02	97,86	98,60
	E	89,12	93,23	95,81	97,67	98,41
Kernel	A	72,18	76,07	78,07	79,05	79,38
	B	62,48	67,61	70,69	72,42	73,22
	C	82,79	89,13	92,53	94,74	95,69
	D	83,99	91,39	95,54	98,11	99,07
	E	83,60	91,23	95,36	98,05	99,01
Exponential	A	79,66	82,59	84,22	84,94	85,24
	B	72,77	76,74	79,28	81,30	82,26
	C	84,44	88,63	91,15	92,93	93,59
	D	87,62	93,06	96,11	98,19	98,99
	E	87,51	92,94	96,02	98,10	98,91

Allokaatiomallien vertailu

Normaalijakauma



Ydinestimointi



EkspONENTTijakauma

Allokaatiomallien vertailu: yhteenveto

- Allokaatiomalleilla A ja B päädyttiin selvästi alhaisimpiin muistin käyttöasteisiin
- D ja E mallit olivat tehokkaimpia
 - Odotettu tulos, sillä mallit pyrkivät minimoimaan palvelimien tyhjäksi jäävän tilan
- Samat mallit suoriutuivat parhaiten kaikilla jakaumilla
 - Erot korostuivat ydinestimonnissa johtuen suurista ympäristöistä

Lähteet

- J. Dugmore, *Capacity Management*, BSI British Standards Institution, 2005.
- N. Sabharwal, P. Wali, *Cloud Capacity Management*, Apress Berkeley, 2013.
- J. Allspaw, *The Art of Capacity Planning*. O'Reilly Media, Inc., 2008.
- T. P. Roseline, C. J. M. Tauro, M. Miranda, *An approach for efficient capacity management in a cloud*, IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), 2017.
- D. Gmach, J. Rolia, L. Cherkasova, A. Kemper, *Capacity management and demand prediction for next generation data centers*, IEEE International Conference on Web Services (ICWS), 2007.
- C. Martínez-Costa, M. Mas-Machuca, E. Benedito, A. Corominas, *A review of mathematical programming models for strategic capacity planning in manufacturing*, International Journal of Production Economics, vol. 153, pp. 66–85, 2014.

Lähteet

- L. V. Kantorovich, *Mathematical methods of organizing and planning production*, Management Science, vol. 6, no. 4, pp. 366–422, 1960.
- P. C. Gilmore, R. E. Gomory, *A linear programming approach to the cutting-stock problem*, Operations Research, vol. 9, no. 6, pp. 849–859, 1961.
- J. Desrosiers, M. E. Lübbecke, *A Primer in Column Generation*, Springer US, 2005.
- M. Rosenblatt, *Remarks on some nonparametric estimates of a density function*, The Annals of Mathematical Statistics, vol. 27, no. 3, pp. 832–837, 1956.