Marja Tuominen
30.1.2000
Avoimien hajautettujen järjestelmien seminaari
Tietojenkäsittelytieteen laitos
Helsingin yliopisto

VIKASIETOISUUS REAALIAIKAISISSA HAJAUTETUISSA JÄRJESTELMISSÄ

Reaaliaikajärjestelmä on kuten mikä tahansa tietokonejärjestelmä, joka saa palvelupyyntöjä, joihin se reagoi. Vasteajaksi kutsutaan aikaa, joka kuluu pyynnön lähettämisestä vastauksen saamiseen. Järjestelmä on yleisesti ymmärrettynä *reaaliaikainen*, jos vasteaika pystytään pitämään alle sellaisen rajan, jonka ylittäminen aiheuttaa järjestelmässä vakavia seuraamuksia. Se mikä luokitellaan vakavaksi seuraamukseksi vaihtelee järjestelmästä ja sen toimintatarkoituksesta riippuen.

Reaaliaikaisuuden saavuttamiseksi on olemassa erilaisia mekanismeja kuten keskusmuistipohjaisuus, rinnakkaisuus ja nopeat tiedonsiirtomenetelmät. Keskusmuistipohjaisuus tarkoittaa sitä, että käsiteltävät tiedot sijaitsevat tietokoneen keskusmuistissa eivätkä jollain hitaalla ulkoisella tietovälineellä. Järjestelmässä voi olla sekä laitteisto- että ohjelmistotasolla monistettuja komponentteja, jotka suorittavat samoja tehtäviä. Näin järjestelmään saadaan rinnakkaisuutta ja järjestelmän saamia palvelupyyntöjä voidaan toteuttaa nopeammin.

Vikasietoisuus tarkoittaa järjestelmän kykyä jatkaa toimintaansa laitteisto- tai ohjelmistovirheestä huolimatta. Vian ilmetessä vikasietoinen järjestelmä joko epäonnistuu tehtävässään hallitusti tai naamioi vian niin, ettei se näy käyttäjälle. Hajautettu järjestelmä koostuu kokoelmasta prosesseja, jotka toimivat samanaikaisesti eri tietokoneilla ja jotka kommunikoivat usein varsin hitaiden ja epäluotettavien tietoliikenneyhteyksien avulla. Pyydetyn palvelun onnistuminen riippuu mahdollisesti usean eri komponentin suorituksen onnistumisesta. Vian sattuminen palvelun suorituksen aikana tällaisessa järjestelmässä on siis todennäköisempää kuin yksittäisen tietokoneen sisällä.

Järjestelmässä ilmenevät viat voidaan karkeasti jaotella laitteisto- ja ohjelmistovikoihin. Laitteistovirheisiin voidaan varautua monistamalla kriittisiä komponentteja. Tällaisia varmennusmenettelyä ovat mm. 2N ja N+1 varmennukset. 2N -menetelmässä kullakin yksittäisellä resurssilla on tarkoin nimetty vararesurssi. N+1 –varmennuksessa

varayksiköitä voi olla yksi tai useampia. Ohjelmistovirheet voidaan puolestaan piilottaa käyttämällä erilaisia toipumisalgoritmeja. Tämä tarkoittaa sitä, että ohjelmisto täytyy suunnitella niin, että vian sattuessa järjestelmän pysyvien tietoalkioiden tila voidaan palauttaa ennalleen.

Korkea vikasietoisuus on sinällään erittäin tavoiteltava ominaisuus. Kuitenkin luotettavuuden kasvattaminen tarkoittaa yleensä myös järjestelmän monimutkaisuuden ja vasteaikojen kasvamista, mikä ei ole tavoiteltavaa, etenkään reaaliaikaisissa järjestelmissä. Perinteisesti tätä ongelmaa on lähestytty niin, että järjestelmälle asetetaan jo suunnitteluvaiheessa vaatimukset eri palvelujen vasteajoista sekä rajataan tarkasti, minkälaisista virheistä järjestelmän tulee pystyä toipumaan. Tämä johtaa kuitenkin tarpeettomaan jäykkyyteen ja ehkä jopa liiankin tiukkaan vikasietoisuuspolitiikkaan joidenkin järjestelmän tarjoamien palvelujen osalta. Parempi ratkaisu onkin *sopeutuva vikasietoisuus* (adaptive fault-tolerance). Koska hajautetut reaaliaikaiset järjestelmät toimivat usein nopeasti vaihtuvassa ympäristössä, myös vikasietoisuusvaatimukset muuttuvat. Niinpä järjestelmän pitäisikin pystyä mukautumaan näihin uusiin vaatimuksiin ja tarjota erilaisia ratkaisuja eri tilanteisiin.

Lähteet

[CoD98] Coulouris G., Dollimore J., Kindberg T., *Distributed systems concepts and design*. Addison Wesley, England 1998.

[Cri91] Cristian F., Understanding Fault-Tolerant Distributed Systems. *Communications of ACM*, vol. 34, no 2, 1991.

[Kop94] Kopetz H., *The Design of Fault-Tolerant Real-Time Systems*. Proceedings of the 20th EUROMICRO Conference on System Architecture and Integration, 1994, Page(s): 4 –9.

[GGL93] Goldberg J., Greenberg I., Lawrence T.F., *Adaptive Fault Tolerance*. Proceedings of the IEEE Workshop on Advances in Parallel and Distributed Systems, 1993, Page(s): 127 –132.