DAFTAR PUSTAKA

[1] M. Agung and A. I. Kistijantoro, “High Performance CDR Processing with MapReduce,” *2015 9th Int. Conf. Telecommun. Syst. Serv. Appl.*, pp. 1–6, 2015.

[2] K. Ahsan, “A Novel Three-layer Architecture for Information System Integration,” *FASSI 2015 First Int. Conf. Fundam. Adv. Softw. Syst. Integr.*, no. c, pp. 7–16, 2015.

[3] S. P. Ahuja, R. Eggen, and A. K. Jha, “A Performance Evaluation of Distributed Algorithms on Shared Memory and *Message* Passing Middleware Platforms,” *Informatica*, vol. 29, pp. 327–333, 2005.

[4] S. S. Alwakeel and H. M. Almansour, “Modeling and performance evaluation of *message*-oriented middleware with priority queuing,” *Information Technology Journal*, vol. 10, no. 1. pp. 61–70, 2011.

[5] S. Chintala, Y. Soussov, and R. C. Tan, “Implementing Thread Priorities in a HTTP Server,” *MULTICORE Program. SPRING 2011 COURSE Proj.*, 2011.

[6] F. S. De Boer, I. Grabe, M. M. Jaghoori, A. Stam, and W. Yi, “Modeling and analysis of thread-pools in an industrial communication platform,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5885 LNCS, pp. 367–386, 2009.

[7] Y. Ding, K. Hu, K. Wu, and Z. Zhao, “Performance Monitoring and Analysis of Task-Based OpenMP,” *PLoS One*, vol. 8, no. 10, p. e77742, 2013.

[8] T. Gee, 2011, " Dissecting the Disruptor: Why it's so fast (part one) - Locks Are Bad", http://mechanitis.blogspot.co.id/2011/07/dissecting-disruptor-why-its-so-fast.html (diakses 30 Maret 2017)

[9] T. Gee, 2011, "Dissecting the Disruptor: Why it's so fast (part two) - Magic cache line padding", http://mechanitis.blogspot.co.id/2011/07/dissecting-disruptor-why-its-so-fast_22.html (diakses 30 Maret 2017)

[10] T. Gee, 2011, "Dissecting the Disruptor: What's so special about a ring buffer?", http://mechanitis.blogspot.co.id/2011/06/dissecting-disruptor-whats-so-special.html (diakses 30 Maret 2017)

[11] T. Gee, 2011, "Dissecting the Disruptor: How do I read from the ring buffer?", http://mechanitis.blogspot.co.id/2011/06/dissecting-disruptor-how-do-i-read-from.html (diakses 30 Maret 2017)

[12] T. Gee, 2011, "Dissecting the Disruptor: Writing to the ring buffer", http://mechanitis.blogspot.co.id/2011/07/dissecting-disruptor-writing-to-

ring.html (diakses 30 Maret 2017)

[13] G. Hohpe and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. The Addison-Wesley, 2003.

[14] R. Maani, S. Camorlinga, and N. Arnason, "A parallel method to improve medical image transmission," *J. Digit. Imaging*, vol. 25, no. 1, pp. 101–109, 2012.

[15] P. Maresca, 2016, "Scalable I/O: Events- Vs Multithreading-based", https://thetechsolo.wordpress.com/2016/02/29/scalable-io-events-vs-multithreading-based/ (diakses 30 Maret 2017)

[16] F. McSherry, M. Isard, and D. G. Murray, "Scalability! But at what COST?," *15th Work. Hot Top. Oper. Syst. (HotOS XV)*, pp. 1–10, 2015.

[17] H. K. Mhatre and K. N. Kanere, "Real Time Messaging Server Using Duplex Web Services," *Int. J. Adv. Comput. Res.*, vol. 4, no. 3, pp. 866–871, 2014.

[18] K. Sachs, S. Kounev, S. Appel, and A. Buchmann, "Benchmarking of *message*-oriented middleware," *Proc. Third ACM Int. Conf. Distrib. Event-Based Syst.*, pp. 1–2, 2009.

[19] K. Sachs, S. Kounev, J. Bacon, and A. Buchmann, "Performance evaluation of *message*-oriented middleware using the SPECjms2007 benchmark," *Perform. Eval. 66*, vol. 66, no. 8, pp. 410–434, 2009.

[20] TimeSys, "The Concise Handbook of Real Time Systems", *TimeSys Corporation*, version 1.3, 2002

[21] M. Thompson, D. Farley, M. Barker, P. Gee, and A. Stewart, "Disruptor: High performance alternative to bounded queues for exchanging data between concurrent threads," 2011.

[22] C. Vinte, "Upon a *Message*-Oriented Trading API," *Inform. Econ. J.*, vol. 14, no. 1, pp. 208–216, 2010.

[23] C. Vinte and C. Vinţe, "Software Architecture Coupling Metric for Assessing Operational Responsiveness of Trading Systems," *Inform. Econ.*, vol. 16, no. 4, pp. 105–116, 2012.

[24] D. Xu and B. Bode, "Performance Study and Dynamic Optimizatoin Design for Thread Pool Systems," *Proc. Int. Conf. Comput. Commun. Control Technol. CCCT'2004*, 2004.