References

[1] Itrs (international technology roadmap for semiconductors). *In: International SEMATECH Austin, TX (http://public.itrs.net)*, 2011.

[2] R. Gallagher A. Parekh. A generalized processor sharing approach to flow-control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, 1993.

[3] L. Abeni. Server mechanisms for multimedia applications. Technical report.

[4] L. Abeni and G.C. Buttazzo. Integrating multimedia applications in hard real-time systems. In *19th IEEE Real-Time Systems Symposium*, pages 4–13, 1998.

[5] P. Altenbernd. Deadline-monotonic software scheduling for the co-synthesis of parallel hard real-time systems. In *EDTC '95: 1995 European conference on Design and Test*, page 190, Washington, DC, USA, 1995. IEEE Computer Society.

[6] J Anderson, V. Bud, and U. C. Devi. An edf-based scheduling algorithm for multiprocessor soft real-time systems. In *17th Euromicro Conference on Real-Time Systems (ECRTS'05)*, pages 199–208, Washington, DC, USA, 2005. IEEE Computer Society.

[7] J. Anderson and A. Srinivasan. Early-release fair scheduling. In *12th Euromicro Conference on Real-Time Systems*, pages 35–43, Jun 2000.

[8] J. Anderson and A. Srinivasan. Mixed pfair/erfair scheduling of asynchronous periodic tasks. *Journal of Computer and System Sciences*, 68(1):157–204, Feb 2004.

[9] J. H. Anderson, J. M. Calandrino, and U. C. Devi. Real-time scheduling on multicore platforms. *12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06)*, 0:179–190, 2006.

[10] B. Andersson and K. Bletsas. Sporadic multiprocessor scheduling with few preemptions. In *2008 Euromicro Conference on Real-Time Systems*, pages 243–252, Washington, DC, USA, 2008. IEEE Computer Society.

[11] B. Andersson and J. Jonsson. The utilization bounds of partitioned and pfair static-priority scheduling on multiprocessors are 50%. In *15th Euromicro Conference on Real-Time Systems*, pages 33–40, Jul 2003.

[12] B. Andersson and E. Tovar. Multiprocessor scheduling with few preemptions. In *12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 322–334, 2006.

[13] B. Andersson and E. Tovar. Multiprocessor scheduling with few preemptions. In *RTCSA '06: 12th IEEE International Conference on Embedded and Real-Time Computing*, pages 322–334, Washington, DC, USA, 2006. IEEE Computer Society.

[14] K. Bletsasand B. Andersson. Notional processors: An approach for multiprocessor scheduling. In *2009 15th IEEE Symposium on Real-Time and Embedded Technology and Applications*, RTAS '09, pages 3–12, Washington, DC, USA, 2009. IEEE Computer Society.

[15] J. Augustine, S. Irani, and C. Swamy. Optimal power-down strategies. In *45th Symp. Foundations of Computer Science (FOCS'04)*, pages 530–539, 2004.

[16] H. Aydin. Exact fault-sensitive feasibility analysis of real-time tasks. *IEEE Transactions on Computers*, 56(10):1372–1386, 2007.

[17] H. Aydin, R. Melhem, D. Mossé, and P. Mejía-Alvarez. Determining optimal processor speeds for periodic real-time tasks with different power characteristics. In *ECRTS '01: 13th Euromicro Conference on Real-Time Systems*, page 225, Washington, DC, USA, 2001. IEEE Computer Society.

[18] H. Aydin, R. Melhem, D. Mossé, and P. Mejía-Alvarez. Determining optimal processor speeds for periodic real-time tasks with different power characteristics. In *ECRTS '01: 13th Euromicro Conference on Real-Time Systems*, page 225, Washington, DC, USA, 2001. IEEE Computer Society.

[19] S. Banachowski and S.A. Brandt. Toward a taxonomy of time-constrained applications. In *Work in Progress 24th IEEE Real-Time Systems Symposium*, pages 3–6, 2003.

[20] P. Baptiste. Scheduling unit tasks to minimize the number of idle periods: A polynomial time algorithm for offline dynamic power management. In *17th annual ACM-SIAM symposium on Discrete algorithm (SODA)*, pages 364–367, New York, NY, USA, 2006. ACM.

[21] P. Baptiste, M. Chrobak, and C. Durr. Polynomial time algorithms for minimum energy scheduling. In *15th Annual European Symposium on Algorithms (ESA)*, pages 136–150, 2007.

[22] S. Baruah, N. Cohen, C.G. Plaxton, and D. Varvel. Proportionate progress: A notion of fairness in resource allocation. *Algorithmica*, 15(6):600–625, 1996.

[23] S. Baruah, J. Gehrke, and C.G. Plaxton. Fast scheduling of periodic tasks on multiple resources. In *9th International Parallel Processing Symposium*, pages 280–288, Apr 1995.

[24] S.K. Baruah, J. Gehrke, C.G. Plaxton, I. Stoica, H.M. Abdel-Wahab, and K. Jeffay. Fair on-line scheduling of a dynamic set of tasks on a single resource. *Information Processing Letters*, 64(1):43–51, 1997.

[25] M. A. Bender, R. Clifford, and K. Tsichlas. Scheduling algorithms for procrastinators. *Journal of Scheduling*, 11(2):95–104, 2008.

[26] E. Berg and E. Hagersten. Statcache: a probabilistic approach to efficient and accurate data locality analysis. In *ISPASS '04: 2004 IEEE International Symposium on Performance Analysis of Systems and Software*, pages 20–27, Washington, DC, USA, 2004. IEEE Computer Society.

[27] K. Bletsas and B. Andersson. Preemption-light multiprocessor scheduling of sporadic tasks with high utilisation bound. In *2009 30th IEEE Real-Time Systems Symposium*, RTSS '09, pages 447–456, Washington, DC, USA, 2009. IEEE Computer Society.

[28] R. Bryant and B. Hartner. Java technology, threads, and scheduling in linux. In *IBM developerWorks Library Paper, IBM Linux Technology Center*, Jan 2000.

[29] A. Burns, R. Davis, and S. Punnekkat. Feasibility analysis of fault-tolerant real-time task sets. In *8th Euromicro Workshop on Real-Time Systems*, pages 29–33, Washington, DC, USA, 1996. IEEE Computer Society.

[30] G. Buttazzo and L. Abeni. Adaptive rate control through elastic scheduling. In *39th IEEE Conference on Decison and Control, Sydney, Australia, vol. 5*, pages 4883–4888, 2000.

[31] G. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni. Elastic scheduling for flexible workload management. *IEEE Transactions on Computers*, 51(3):289–302, 2002.

[32] G.C. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Kluwer, 1997.

[33] J. Carlson, T. Lennvall, and G. Fohler. Value based overload handling of aperiodic tasks in offline scheduled real-time systems. In *Work-in-progress Session, 13th Euromicro Conference on Real-Time Systems, Delft, The Netherlands*, 2001.

[34] John Carpenter, Shelby Funk, Philip Holman, Anand Srinivasan, James Anderson, and Sanjoy Baruah. A categorization of real-time multiprocessor scheduling problems and algorithms, 2004.

[35] J. Chen and T. Kuo. Procrastination for leakage-aware rate-monotonic scheduling on a dynamic voltage scaling processor. In *ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*, pages 153–162, 2006.

[36] J. Chen and T. Kuo. Procrastination determination for periodic real-time tasks in leakage-aware dynamic voltage scaling systems. In *ICCAD '07: 2007 IEEE/ACM international conference on Computer-aided design*, pages 289–294, Piscataway, NJ, USA, 2007. IEEE Press.

[37] H. Cho, B. Ravindran, and E. Jensen. An optimal real-time scheduling algorithm for multiprocessors. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 101–110, 2006.

[38] K. Choi, G. Jung, T. Kim, and S.Jung. Real-time scheduling algorithm for minimizing maximum weighted error with $o(n \lg n + cn)$ complexity. *Information Processing Letters*, 67:311–315, 1998.

[39] M. Chtepen, F.H.A. Claeys, B. Dhoedt, F. Turck, P. Demeester, and P.A. Vanrolleghem. Adaptive task checkpointing and replication: Toward efficient fault-tolerant grids. *IEEE Transactions on Parallel and Distributed Systems*, 20(2):180–190, 2009.

[40] R.W. Conway, W.L. Maxwell, and L.W. Miller. *Theory of Scheduling*. Dover Publications, Inc., 31 East 2nd Street, Mineola, N.Y. 11501, 2003.

[41] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM '89, Austin, TX*, pages 1–12, Sep 1989.

[42] Z. Deng and J.W.S. Liu. Scheduling real-time applications in open environment. In *18th IEEE Real-Time Systems Symposium*, pages 308–319, Dec 1997.

[43] Z. Deng, J.W.S. Liu, and J. Sun. A scheme for scheduling hard real-time applications in open system environment. In *9th Euromicro Workshop on Real-Time Systems*, pages 191–199, 1997.

[44] S. Dhall and C. Liu. On a real-time scheduling problem. *Operations research*, 26(1):127–140, 1978.

[45] A. Fedorova, M. Seltzer, C. Small, and D. Nussbaum. Performance of multithreaded chip multiprocessors and implications for operating system design. In *USENIX 2005 Annual Technical Conference*, pages 395–398, Apr 2005.

[46] C.J. Fidge. Real-time scheduling theory. Technical Report 02-19, Apr 2002.

[47] S. Gopalakrishnan and X. Liu. Reclaiming spare capacity and improving aperiodic response times in real-time environments. article id 391215. *EURASIP Journal of Embedded Systems*, 2011.

[48] A. Gupta, A. Tucker, and S. Urushibara. The impact of operating system scheduling policies and synchronization methods of the performance of parallel application. In *ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 120–132, 1991.

[49] T. Dohi H. Okamura *. Comprehensive evaluation of aperiodic checkpointing and rejuvenation schemes in operational software system. *Journal of Systems and Software*, 83(9):1591–1604, 2010.

[50] J. Hansson, M. Thuresson, and S.H. Son. Imprecise task scheduling and overload management using or-uld. In *7th International Conference on Real-Time Computing Systems and Applications, Cheju Island, S. Korea*, pages 307–314, 2000.

[51] P. Holman and J. H. Anderson. Group-based pfair scheduling. *Real-Time Systems*, 32:125–168, February 2006.

[52] S. Irani and K. R. Pruhs. Algorithmic problems in power management. *SIGACT News*, 36(2):63–76, 2005.

[53] S. Irani, S. Shukla, and R. Gupta. Online strategies for dynamic power management in systems with multiple power-saving states. *Trans. on Embedded Computing Sys.*, 2(3):325–346, 2003.

[54] S. Irani, S. Shukla, and R. Gupta. Online strategies for dynamic power management in systems with multiple power-saving states. *Trans. on Embedded Computing Sys.*, 2(3):325–346, 2003.

[55] K. Jeffay. Scheduling sporadic tasks with shared resources in hard real-time systems. In *13th IEEE Real-Time Systems Symposium*, pages 89–98, Dec 1992.

[56] K. Jeffay and D. Bennett. A rate-based execution abstraction for multimedia computing. In *Network and Operating System Support for Digital Audio and Video*, pages 64–75, 1995.

[57] K. Jeffay and S. Goddard. A theory of rate-based execution. In *20th IEEE Real-Time Systems Symposium*, pages 304–314, 1999.

[58] K. Jeffay and S. Goddard. Rate-based resource allocation models for embedded systems. *Lecture Notes in Computer Science*, 2211:204–, 2001.

[59] K. Jeffay and G. Lamastra. A comparative study of the realization of rate-based computing services in general purpose operating systems. In *7th IEEE International Conference on Real-Time Computing Systems and Applications*, pages 81–90, Dec 2000.

[60] R. Jejurikar and R. Gupta. Procrastination scheduling in fixed priority real-time systems. *ACM SIGPLAN Notices*, 39(7):57–66, 2004.

[61] R. Jejurikar and R. Gupta. Dynamic slack reclamation with procrastination scheduling in real-time embedded systems. In *DAC '05: 42nd annual conference on Design automation*, pages 111–116, New York, NY, USA, 2005. ACM.

[62] R. Jejurikar, C. Pereira, and R. Gupta. Leakage aware dynamic voltage scaling for real-time embedded systems. In *DAC '04: 41st annual conference on Design automation*, pages 275–280, New York, NY, USA, 2004. ACM.

[63] R. Jurgen. *Electronic Braking, Traction, and Stability Controls, Vol 2*. SAE Technical Paper PT-129, 2006.

[64] S. Kato and N. Yamasaki. Real-time scheduling with task splitting on multiprocessors. In *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, RTCSA '07, pages 441–450, Washington, DC, USA, 2007. IEEE Computer Society.

[65] S. Kato and N. Yamasaki. Portioned edf-based scheduling on multiprocessors. In *8th ACM international conference on Embedded software*, EMSOFT '08, pages 139–148, New York, NY, USA, 2008. ACM.

[66] K.Funaoka, S. Kato, and N. Yamasaki. Work-conserving optimal real-time scheduling on multiprocessors. In *20th Euromicro Conference on Real-Time Systems (ECRTS'08)*, pages 13–22, Washington, DC, USA, 2008. IEEE Computer Society.

[67] N.S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J.S. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan. Leakage current: Moore's law meets static power. *Computer*, 36:68–75, 2003.

[68] S. Kim, D. Chandra, and Y. Solihin. Fair cache sharing and partitioning on a chip multiprocessor architecture. In *13th International Conference on Parallel Architecture and Compilation Techniques*, pages 111–122, 2004.

[69] T. Kimbrel, B. Schieber, and M. Sviridenko. Minimizing migrations in fair multiprocessor scheduling of persistent tasks. *Journal of Scheduling*, 9(4):365–379, Aug 2006.

[70] K. Klonowska, L. Lundberg, and H. Lennerstad. The maximum gain of increasing the number of preemptions in multiprocessor scheduling. *Acta Inf.*, 46:285–295, June 2009.

[71] S.W. Kwak, B.J. Choi, and B.K. Kim. An optimal checkpointing-strategy for real-time control systems under transient faults. *IEEE Transactions on Reliability*, 50(3):293–301, 2001.

[72] H. Kweon, Y. Do, J. Lee, and B. Ahn. An efficient power-aware scheduling algorithm in real time system. In *PacRim '07: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 350–353, Aug 2007.

[73] P. Langen and B. Juurlink. Leakage-aware multiprocessor scheduling. *Journal of Signal Processing Systems*, 57:73–88, October 2009.

[74] H. Lee, H. Shin, and S. Min. Worst case timing requirement of real-time tasks with time redundancy. In *6th International Conference on Real-Time Computing Systems and Applications*, pages 410–414, Washington, DC, USA, 1999. IEEE Computer Society.

[75] I. Lee, J. Leung, and S. H. Son, editors. *Handbook of Real-Time and Embedded Systems*. Chapman & Hall/CRC Press, 2007.

[76] Y. Lee, K. P. Reddy, and C. M. Krishna. Scheduling techniques for reducing leakage power in hard real-time systems. In *15th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 105–112, 2003.

[77] J. Leung. A new algorithm for scheduling periodic, real-time tasks. *Algorithmica*, 4(1):209–219, 1989.

[78] G. Levin, S. Funk, C. Sadowski, I. Pye, and S. Brandt. Dp-fair: A simple model for understanding optimal multiprocessor scheduling. In *22nd Euromicro Conference on Real-Time Systems*, pages 3–13, Washington, DC, USA, 2010. IEEE Computer Society.

[79] K.J. Lin, S. Natarajan, and J.W.S. Liu. Imprecise results: utilizing partial computations in real-time systems. In *RTSS '87: 8th Real-Time Systems Symposium*, pages 210–217, 1987.

[80] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1):46–61, 1973.

[81] J. W. S. Liu. *Real-Time Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[82] X. Liu and S. Goddard. Supporting dynamic qos in linux. In *10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'04)*, page 246, Washington, DC, USA, 2004. IEEE Computer Society.

[83] X. Liu and S. Goddard. Scheduling legacy multimedia applications. *Journal of Systems and Software*, 75(3):319–328, Mar 2005.

[84] J. M. Lopez, M. García, J. L. Diaz, and D. F. Garcia. Utilization bounds for multiprocessor rate-monotonic scheduling. *Real-Time Systems Journal*, 24:5–28, 2003.

[85] J.M. Lopez, M. Garcia, J.L. Diaz, and D.F. Garcia. Worst-case utilization bound for edf scheduling on real-time multiprocessor systems. In *12th Euromicro Conference on Real-Time Systems*, pages 25–33, Jun 2000.

[86] Z. Lu, Y. Zhang, M. Stan, J. Lach, and K. Skadron. Procrastinating voltage scheduling with discrete frequency sets. In *DATE '06: conference on Design, automation and test in Europe*, pages 456–461, 3001 Leuven, Belgium, Belgium, 2006. European Design and Automation Association.

[87] P. Lun. Report on simple one-dimensional on-line bin packing algorithms. `http://www.tcs.fudan.edu.cn/rudolf/Courses/Online/Online03/po_report.pdf`.

[88] J. Malkevitch. Bin packing and machine scheduling. *Feature Column from the AMS: Monthly Essays on Mathematical Topics*, Jun 2004.

[89] P. Mejia-Alvarez, R. Melhem, D. Mosse, and H. Aydin. An incremental server for scheduling overloaded real-time systems. *IEEE Transactions on Computers*, 52(10):1347–1361, 2003.

[90] A. Merchand and M. Chetto. Dynamic scheduling of periodic skippable tasks in an overloaded real-time system. In *AICCSA 2008: IEEE/ACS International conference on Computer Systems and Applications*, pages 456–464, 2008.

[91] A. K. Mok. Fundamental design problems of distributed systems for the hard-real-time environment. Technical report, Cambridge, MA, USA, 1983.

[92] N. Naksinehaboon, M. Paun, R. Nassar, B. Leangsuksun, and S. Scott. High performance computing systems with various checkpointing schemes. *International Journal of Computers, Communications & Control*, 4(4):386–400, 2009.

[93] H.S. Negi, T.Mitra, and A. Roychoudhury. Accurate estimation of cache-related preemption delay. In *CODES+ISSS '03: 1st IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 201–206, 2003.

[94] J. Nieh, C. Vaill, and H. Zhong. Virtual-time round-robin: An o(1) proportional share scheduler. In *USENIX Annual Technical Conference*, pages 245–259, Jun 2001.

[95] J. Nieh, C. Vaill, and H. Zhong. Group ratio round-robin: An O(1) proportional share scheduler. In *General Track: 2004 USENIX Annual Technical Conference*, pages 245–259, Jun 2004.

[96] D. Niz and R. Rajkumar. Partitioning bin-packing algorithms for distributed real-time systems. *International Journal of Embedded Systems*, 2(3):196–208, 2006.

[97] D. Oh and T. P. Bakker. Utilization bounds for n-processor rate monotonescheduling with static processor assignment. *Real-Time Systems Journal.*, 15:183–192, 1998.

[98] A. Oliner, L. Rudolph, and R. Sahoo. Cooperative checkpointing theory. In *20th International Parallel and Distributed Processing Symposium (IPDPS '06)*, pages 132–132, 2006.

[99] A. Oliner and R. Sahoo. Evaluating cooperative checkpointing for supercomputing systems. In *20th International Parallel and Distributed Processing Symposium (IPDPS '06)*, pages 363–363, 2006.

[100] J.S. Plank, Y. Chen, K. Li, M. Beck, and G. Kingsley. Memory exclusion: Optimizing the performance of checkpointing systems. *Software: Practice and Experience*, 29(2):125–142, 1999.

[101] S. Punnekkat, A. Burns, and R. Davis. Analysis of checkpointing for real-time systems. *International Journal of Time-Critical Computing Systems*, 20(1):83–102, 2001.

[102] S. Ramabhadran and J. Pasquale. Stratified round robin: A low complexity packet scheduler with bandwidth fairness and bounded delay. In *ACM SIGCOMM*, pages 239–249, 2003.

[103] P. Ramanathan and M. Hamdaoui. A dynamic priority assignment technique for streams with (m, k)-firm deadlines. *IEEE Transaction on Computers*, 44(12):1443–1451, 1995.

[104] J. Regehr, M. Jones, and J. Stankovic. Operating system support for multimedia: The programming model matters. Technical Report Microsoft Research (MSR-TR-2000-89), Sep 2000.

[105] A. Sarkar, P. P. Chakrabarti, and S. Ghose. Partition oriented frame based fair scheduler. *Journal of Parallel Distributed Computing*, 70(7):707–718, 2010.

[106] A. Sarkar, P.P. Chakrabarti, and R. Kumar. Frame-based proportional round-robin. *IEEE Transactions on Computers*, 55(9):1121–1129, 2006.

[107] A. Sarkar, S. Swaroop, S. Ghose, and P. P. Chakrabarti. Erfair scheduler with processor shutdown. In *16th International Conference on High Performance Computing (HiPC '09)*, pages 4–12, 2009.

[108] Y. Shin, K. Choi, and T. Sakurai. Power optimization of real-time embedded systems on variable speed processors. In *ICCAD '00: 2000 IEEE/ACM international conference on Computer-aided design*, pages 365–368, Piscataway, NJ, USA, 2000. IEEE Press.

[109] Y. Shin, K. Choi, and T. Sakurai. Power optimization of real-time embedded systems on variable speed processors. In *ICCAD '00: 2000 IEEE/ACM international conference on Computer-aided design*, pages 365–368, Piscataway, NJ, USA, 2000. IEEE Press.

[110] T. Simunic, L. Benini, P.W. Glynn, and G. De Micheli. Dynamic power management for portable systems. In *6th annual international conference on Mobile Computing and Networking*, pages 11–19, 2000.

[111] M. Spuri and G.C. Buttazzo. Efficient aperiodic service under earliest deadline scheduling. In *15th IEEE Real-Time Systems Symposium*, pages 2–11, Dec 1994.

[112] M. Spuri, G.C. Buttazzo, and F. Sensini. Robust aperiodic scheduling under dynamic priority systems. In *16th IEEE Real-Time Systems Symposium*, pages 210–221, 1995.

[113] M. S. Squillante and E. D. Lazowska. Using processor-cache affinity information in shared-memory multiprocessor scheduling. *IEEE Transactions on Parallel and Distributed Systems*, 4(2):131 – 143, 1993.

[114] M.S. Squillante and R.D. Nelson. Analysis of task migration in shared-memory multiprocessor scheduling. In *ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 143–155, 1991.

[115] A. Srinivasan, P. Holman, and J. Anderson. The case for fair multiprocessor scheduling. In *11th International Workshop on Parallel and Distributed Real-time Systems, Nice, France*, Apr 2003.

[116] A. Srinivasan, P. Holman, and J. H. Anderson. Integrating aperiodic and recurrent tasks on fair-scheduled multiprocessors. In *14th Euromicro Conference on Real-Time Systems*, 2002.

[117] J. Stankovic. Misconceptions about real-time computing: A serious problem for next generation systems. *IEEE Computer*, 28(6):10–19, Oct 1988.

[118] J.A. Stankovic. Continuous and multimedia os support in real-time control applications. In *HOTOS '95: Fifth Workshop on Hot Topics in Operating Systems (HotOS-V)*, page 8, Washington, DC, USA, 1995. IEEE Computer Society.

[119] I. Stoica, H. Abdel-Wahab, and K. Jeffay. On the duality between resource reservation and proportional share resource allocation. Technical Report TR_96_19, 1996.

[120] I. Stoica, H. Abdel-Wahab, K.Jeffay, S.Baruah, J. Gehrke, and C.Plaxton. A proportional share resource allocation algorithm for real-time, time-shared systems. In *17th IEEE Real-Time Systems Symposium*, pages 288–299, Dec 1996.

[121] B.R. Swim, M. Tayli, M. Benmaiza, and M.C. Woodward. Avoiding deadline decay under transient overloads. In *3rd workshop on Parallel and Distributed Real-Time Systems*, pages 198–200, 1995.

[122] J. Torrellas, A. Tucker, and A. Gupta. Benefits of cache-affinity scheduling in shared-memory multiprocessors: a summary. *SIGMETRICS Perform. Eval. Rev.*, 21(1):272–274, 1993.

[123] C. A. Waldspurger. Lottery and stride scheduling: Flexible proportional-share resource management. Technical Report MIT/LCS/TR-667, 1995.

[124] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, and P. Stenstrom. The worst-case execution-time problem—overview of methods and survey of tools. *ACM Transactions on Embedded Computing Systems (TECS)*, 7(3):1–53, 2008.

[125] M. Wu and M. Shih. Simulated and experimental study of hydraulic anti-lock braking system using sliding-mode pwm control. *Mechatronics*, 13(4):331–351, 2003.

[126] Y. Xiang, Z. Li, and H. Chen. Optimizing adaptive checkpointing schemes for grid workflow systems. In *5th International Conference on Grid and Cooperative Computing Workshops (GCCW'06)*, pages 181–188, 2006.

[127] Y. Zhang and K. Chakrabarty. Fault recovery based on checkpointing for hard real-time embedded systems. In *18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 320–327, Washington, DC, USA, 2003. IEEE Computer Society.

[128] D. Zhu, D. Moss´e, and R. Melhem. Multiple-resource periodic scheduling problem: how much fairness is necessary? In *IEEE Real-Time Systems Symposium*, pages 142–151, Dec 2003.

[129] A. Ziv and J. Bruck. An on-line algorithm for checkpoint placement. *IEEE Transactions on Computers*, 46(9):976–985, 1997.