REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
20-12-2013	Final Report	15-Sep-2009 - 14-Sep-2013

4. TITLE AND SUBTITLE A Real-Time Linux for Multicore Platforms	5a. CONTRACT NUMBER W911NF-09-1-0535
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 611102
6. AUTHORS James H. Anderson	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES University of North Carolina - Chapel Hill Office of Sponsored Research 104 Airport Drive, Suite 2200, CB #1350 Chapel Hill, NC 27599 -1350	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO 11. SPONSOR/MONITOR'S REPORT NUMBER(S) 55837-CS.72

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

The objective of this project is to extend a prototype Linux-based real-time operating system developed previously at UNC (under ARO support) to obtain a fully-functional OS for supporting real-time workloads on multicore platforms. This system, called LITMUS-RT (LInux Testbed for MUltiprocessor Scheduling in Real-Time systems), allows different multiprocessor real-time scheduling and synchronization policies to be specified as plugin components. LITMUS-RT is open-source software (available at

15. SUBJECT TERMS

real-time, Linux, multicore

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			James Anderson
UU	UU	UU	UU		19b. TELEPHONE NUMBER 919-962-1757

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

Report Title

A Real-Time Linux for Multicore Platforms

ABSTRACT

The objective of this project is to extend a prototype Linux-based real-time operating system developed previously at UNC (under ARO support) to obtain a fully-functional OS for supporting real-time workloads on multicore platforms. This system, called LITMUS-RT (LInux Testbed for MUltiprocessor Scheduling in Real-Time systems), allows different multiprocessor real-time scheduling and synchronization policies to be specified as plugin components. LITMUS-RT is open-source software (available at http://www.litmus-rt.org).

The pre-existing LITMUS-RT implementation included plugins for several real-time scheduling and synchronization policies. However, as its name suggests, LITMUS-RT was originally designed as a testbed for empirically comparing such policies. Extending LITMUS-RT so that it can host real applications will require extensions on several fronts. These include: (i) new synchronization and scheduling algorithms that are better suited for large multicore platforms; (ii) new analysis and implementation methods for dealing with non-CPU resources such as graphics processing units (GPUs); (iii) support for dynamic task behavior (tasks can be created, destroyed, and their timing parameters altered at runtime); and (iv) mechanisms for supporting precedence constraints and task suspensions (e.g., due to I/0). The research in this project has led to several joint efforts with colleagues in industry and government labs involving military avionics.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received		Paper
08/23/2011	17.00	Hennadiy Leontyev, Jim Anderson, Bjorn B. Brandenburg. An Overview of Interrupt Accounting Techniques for Multiprocessor Real-Time Systems",, Journal of Systems Architecture, (08 2011): 638. doi:
08/23/2011	18.00	Jim Anderson, Bjorn B. Brandenburg . Spin-Based Reader-Writer Synchronization for Multiprocessor Real-Time Systems", Real-Time Systems, (04 2010): 25. doi:
08/25/2011	37.00	V. Bonifaci, Sanjoy Baruah,, A. Marchetti-Spaccamela, S. Stiller. Improved multiprocessor global schedulability analysis, Real-Time Systems, (03 2010): 3. doi:
12/20/2013	64.00	Cong Liu, James Anderson. Supporting soft real-time parallel applications on multiprocessors, Journal of Systems Architecture, (07 2013): 0. doi:
12/20/2013	65.00	Jeremy P. Erickson, James H. Anderson, Bryan C. Ward. Fair lateness scheduling: reducing maximum lateness in G-EDF-like scheduling, Real-Time Systems, (07 2013): 0. doi: 10.1007/s11241-013-9190-4

TOTAL: 5

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

James H. Anderson, "An Emerging Theory of Soft Real-Time on Multicore," departmental colloquium talk given at Washington University at Saint Louis, October 2013.

James H. Anderson, Panelist, "Safety Critical and Mixed Critical Systems Research: Are We Addressing the Right Challenges?," The International Conference on Embedded Software (EMSOFT), Montreal, Canada, October 2013.

James H. Anderson, Tutorial Organizer (with Lothar Thiele, Madeleine Faugere, and Nikolay Stoimenov), "Mixed-Criticality and Certification," Embedded Systems Week, Montreal, Canada, September-October, 2013.

James H. Anderson, Lecturer, International Summer School on Trends in Computing, Tarragona, Spain, July 2013.

James H. Anderson, "Research Allocation Infrastructure for Real-Time Computing on Multicore Platforms," talk given at the U.S. Army Research Laboratory, Aberdeen Proving Grounds, MD, June 2013.

James H. Anderson, "A Linear Model for Setting Priority Points in Soft Real-Time Systems," talk given at York University, U.K., March 2013.

James H. Anderson, "Supporting UAV Workloads on Multicore Platforms," talk given at Northrop Grumann Corp., Los Angeles, CA, December 2012.

James H. Anderson, "A Multicore Real-Time Mixed-Criticality Framework for Avionics," talk given at the Safe and Secure Systems and Software Symposium (S5), June 2012, in Fairborn, Ohio.

James H. Anderson, Keynote Talk, "Real-Time Multiprocessor Scheduling: Connecting Theory and Practice," 18th International Conference on Real-Time and Network Systems, Toulouse, France, November 2010.

James H. Anderson, "Optimality Results for Multiprocessor Real-Time Locking," colloquium at The University of California at Santa Cruz, July 2011.

James H. Anderson, "Real-Time Computing on Multicore Platforms," tutorial presented at the Uppsala Programming for Multicore Research Center (UPMARC) Summer School, Uppsala, Sweden, June 2010.

Number of Presentations: 11.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
12/20/2013	66.00	Bryan Ward, James Anderson. Multi-Resource Real-Time Reader/Writer Locks for Multiprocessors, 28th IEEE International Parallel and Distributed Processing Symposium. 19-MAY-14, . : ,

TOTAL: 1

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
07/18/2012	54.00	Cong Liu and James H. Anderson, Department of Computer Science, University of North Carolina at Chapel Hill. Supporting Soft Real-Time Parallel Applications on Multicore Processors, 18 Annual IEEE International Conference. , . : ,
07/18/2012	58.00	Jonathan L. Herman,y Christopher J. Kenna,y Malcolm S. Mollison,y James H. Andersony and Daniel M. Johnsonz, yThe University of North Carolina at Chapel Hill. RTOS Support for Multicore Mixed-Criticality Systems, 18th IEEE Real-Time Embedded Tech. a& Appl. Symposium. , . : ,
07/18/2012	57.00	Jeremy P. Erickson and James H. Anderson, University of North Carolina at Chapel Hill. Fair Lateness Scheduling: Reducing Maximum Lateness in G-EDF-likeScheduling, 23rd Euromicro Conference on Real-Time Systems. , . : ,
07/18/2012	56.00	Glenn A. Elliott and James H. Anderson, Department of Computer Science, University of North Carolina at Chapel Hill. Robust Real-Time Multiprocessor Interrupt Handling Motivated by GPUs, 23rd Euromicro COnference on Real-Time Systems. , . : ,
07/18/2012	48.00	Jeremy P. Erickson ,James H. Anderson. Response Time Bounds for G-EDF WithoutIntra-Task Precedence Constraints, 15th International Conference On Principals of Distributed Systems. , . : ,
07/18/2012	55.00	Bryan C. Ward and James H. Anderson, Department of Computer Science, University of North Carolina at Chapel Hill. Supporting Nested Locking in Multiprocessor Real-Time Systems, 23rd Euromicro Conference on Real-Time Systems. , . : ,
07/19/2012	60.00	Jeremy P. Erickson and James H. Anderson, University of North Carolina at Chapel Hill. Response Time Bounds for G-EDF WithoutIntra-Task Precedence Constraints, 15th International COnference On Principals of Distributed Systems. , . : ,
07/19/2012	63.00	Bipasa Chattopadhyay Sanjoy Baruah. Partitioned scheduling of implicit-deadline sporadic task systems under multipleresource constraints, 18th IEEE International Conference on Embedded and Real-Time Computing Systems. , . : ,
07/19/2012	62.00	Sanjoy Baruah, The University of North Carolina. Certification-cognizant scheduling of tasks with pessimistic frequency specification, IEEE Symposium in Industrial Embedded Systems. , . : ,
07/19/2012	61.00	The University of North Carolina at Chapel Hill, Haohan Li Sanjoy Baruah. Global mixed-criticality scheduling onmultiprocessors, 23rd Euromicro Conference on Real-Time Systems. , . : ,
07/19/2012	59.00	Greg Coombe Google, Inc., James H. Anderson University of North Carolina at Chapel Hill, Jeremy P. Erickson University of North Carolina at Chapel Hill. Soft Real-Time Scheduling in Google Earth, 18th IEEE Real-Time and Embedded Tech. and Applications Symposium. , . : ,
08/23/2011	19.00	C. Kenna, J. Herman, B. Brandenburg, A. Mills, J. Anderson. Soft Real-Time on Multiprocessors:Are Analysis-Based Schedulers ReallyWorth It?, 32nd IEEE Real-Time Systems Symposium. , . : ,

08/23/2011 35.00 Sanjoy Baruah,, Gerhard Fohler. Certification-cognizant time-triggered scheduling ofmixed-criticality systems,
IEEE Real-Time Systems Symposium. , . : ,

08/23/2011 34.00 S. Barauh,. The partitioned EDF scheduling of sporadic task systems,
IEEE Real-Time Systems Symposium. , . : ,

08/23/2011 33.00 M. Niemeier, A. Wiese, S. Baruah. Partitioned real-time scheduling on heterogeneousshared-memory multiprocessors,
Euromicro Conference on Real-Time Systems. , , . : ,

08/23/2011 32.00 B. Chattopadhyay, S. Baruah. A lookup-table driven approach to partitioned scheduling└,
IEEE Real-Time Technology and Applications Symposium . , . : ,

08/23/2011 27.00 J.Erickson, J. Anderson. Response Time Bounds for G-EDF Without Intra-Task Precedence Constraints",,
IEEE Real-Time Systems Symposium. , . : ,

08/23/2011 26.00 A. Bastoni, B. Brandenburg, J. Anderson. " Is Semi-Partitioned Scheduling Practical?",
Proceedings of the 23rd Euromicro Conference on Real-Time Systems . , , . : ,

08/23/2011 25.00 M. Mollison, J. Anderson. " Virtual Real-Time Scheduling",
proceedings of the 7th International Workshop on Operating Systems Platforms for Embedded Real-Time Applications. , . : ,

08/23/2011 24.00 C. Liu, J. Anderson. " Supporting Graph-Based Real-Time Applications in Distributed Systems",
proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications. , . : ,

08/23/2011 23.00 A. Mills, J. Anderson. " A Multiprocessor Server-Based Scheduler for Soft Real-Time Tasks with Stochastic Execution Demand",
proceedings of the 17th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications. , . : ,

08/23/2011 22.00 G. Elliott, Jim Anderson. " Real-World Constraints of GPUs in Real-Time Systems",
The first International Workshop on Cyber-Physical Systems. , . : ,

08/23/2011 21.00 B. Brandenburg, J. Anderson. " Real-Time Resource-Sharing under Clustered Scheduling: Mutex, Reader-Writer, and k-Exclusion Locks",,
Proceedings of the International Conf. on Embedded Software. , . : ,

08/23/2011 20.00 G. Elliott, J. Anderson. An Optimal k-Exclusion Real-Time Locking Protocol Motivated by Multi-GPU Systems",
19th International Conference on Real-Time and Newtork Systems . , , . : ,

12/20/2013 67.00 Glenn Elliott, Bryan Ward, James Anderson. GPUSync: A Framework for Real-Time GPU Management,
34th IEEE Real-Time Systems Symposium. 03-DEC-13, . : ,

12/20/2013 68.00 Bryan Ward, James Anderson. Fine-Grained Multiprocessor Real-Time Locking with Improved Blocking,
21st International Conference on Real-Time Networks and Systems. 16-OCT-13, . : ,

12/20/2013 69.00 Bryan Ward, Jonathan Herman, Chris Kenna, James Anderson. Making Shared Caches More Predictable on Multicore Platforms,
25th Euromicro Conference on Real-TimeSystems. 09-JUL-13, . : ,

12/20/2013 70.00 Jeremy Erickson, James Anderson. Reducing Tardiness Under Global Scheduling by Splitting Jobs,
25th Euromicro Conference on Real-TimeSystems. 09-JUL-13, . : ,

12/20/2013 71.00 Cong Liu, James Anderson. Suspension-Aware Analysis for Hard Real-Time Multiprocessor Scheduling,
25th Euromicro Conference onReal-Time Systems. 09-JUL-13, . : ,

TOTAL: **29**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received		Paper
07/18/2012	49.00	Cong Liu and James H. Anderson. An O(m) Analysis Technique for Supporting Real-Time Self-Suspending TaskSystems, ( )
07/18/2012	53.00	Malcolm S. Mollison and James H. Anderson. Bringing Theory Into Practice:A Userspace Library for Multicore Real-Time Scheduling, ( )
07/18/2012	52.00	Bryan C. Ward and James H. Anderson. Nested Multiprocessor Real-Time Locking with Improved Blocking, ( )
07/18/2012	51.00	Christopher J. Kenna, Jonathan L. Herman, Bryan C. Ward, and James H. Anderson. Making Shared Caches More Predictable on Multicore Platforms └, ( )
07/18/2012	50.00	Glenn A. Elliott, Bryan C. Ward, and James H. Anderson. GPUSync: Architecture-Aware Management of GPUs forPredictable Multi-GPU Real-Time Systems, ( )
07/27/2010	2.00	C. Liu and J. Anderson. "Supporting Soft Real-Time DAG-based Systems on Multiprocessors with No Utilization Loss," , (07 2010)
07/27/2010	1.00	B. Brandenburg and J. Anderson. "Optimality Results for Multiprocessor Real-Time Locking," , (07 2010)
07/27/2010	3.00	A. Bastoni, B. Brandenburg, and J. Anderson. "An Empirical Comparison of Global, Partitioned, and Clustered Multiprocessor Real-Time Schedulers," Proceedings of the 31st Real-Time Systems Symposium, San Diego, CA, IEEE Computer Society Press, December 2010, to appear. , (07 2010)
07/27/2010	4.00	C. Liu and J. Anderson, . "Improving the Schedulability of Sporadic Self-Suspending Soft Real-Time Multiprocessor Task Systems," Proceedings of the 16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Macau, China, August 2010, to appear. , (07 2010)
07/27/2010	5.00	A. Bastoni, B. Brandenburg, and J. Anderson, . "Cache-Related Preemption and Migration Delays: Empirical Approximation and Impact on Schedulability," Proceedings of the Sixth International Workshop on Operating Systems Platforms for Embedded Real-Time Applications, Brussels, Belgium, pages 33-44, July 2010. , (07 2010)
07/27/2010	6.00	M. Mollison, J. Erickson, J. Anderson, S. Baruah, and J. Scoredos. "Mixed Criticality Real-Time Scheduling for Multicore Systems," Proceedings of the 7th IEEE International Conference on Embedded Software and Systems, Bradford, U.K., IEEE Computer Society Press, June 2010, to appear. , (07 2010)

07/27/2010 7.00 A. Mills and J. Anderson.
"A Stochastic Framework for Multiprocessor Soft Real-Time Scheduling," Proceedings of the 16th IEEE Real-Time and Embedded Technology and Applications Symposium, Stockholm, Sweden, IEEE Computer Society Press, pages 311-320, April 2010. ,
(07 2010)

07/27/2010 8.00 C. Liu and J. Anderson. "Scheduling Suspendable, Pipelined Tasks with Non-Preemptive Sections in Soft Real-Time Multiprocessor Systems," Proceedings of the 16th IEEE Real-Time and Embedded Technology and Applications Symposium, Stockholm, Sweden, IEEE Computer Society Press, pages 23-32, April 2010.
,
(07 2010)

07/27/2010 9.00 N. Fisher, J. Goossens, and S. Baruah. "Optimal Online Multiprocessor Scheduling of Sporadic Real-Time Tasks is Impossible," Real-Time Systems 45(1), pp 26-71, 2010. ,
(03 2010)

07/27/2010 10.00 S. Baruah, V. Bonifaci, G. D'Angelo, H. Li, A. Marchetti-Spaccamela, N. Megow, and L. Stougie, . "Scheduling Real-Time Mixed-Criticality Jobs," Proceedings of the 35th International Symposium on the Mathematical Foundations of Computer Science (MFCS), Brno, Czech Republic, Springer-Verlag, August 2010, to appear. ,
(07 2010)

07/27/2010 11.00 S. Baruah. "Preemptive Uniprocessor Scheduling of Non-Cyclic GMF Task Systems," Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Macau, China, IEEE Computer Society Press, August 2010, to appear. ,
(07 2010)

07/27/2010 12.00 J. Erickson, U. Devi, and S. Baruah. "Improved tardiness bounds for Global EDF," Proceedings of the EuroMicro Conference on Real-Time Systems, Brussels, Belgium, IEEE Computer Society Press, July 2010. ,
(07 2010)

07/27/2010 13.00 S. Baruah, H. Li, and L. Stougie. "Towards the Design of Certifiable Mixed-Criticality Systems," Proceedings of the IEEE Real-Time Technology and Applications Symposium, Stockholm, Sweden, IEEE Computer Society Press, April 2010. ,
(07 2010)

07/27/2010 14.00 S. Baruah. "An Improved Global EDF Schedulability Test for Uniform Multiprocessors," Proceedings of the IEEE Real-Time Technology and Applications Symposium, Stockholm, Sweden, IEEE Computer Society Press, April 2010.
,
(07 2010)

07/27/2010 15.00 S. Baruah, H. Li, and L. Stougie. "Mixed-Criticality Scheduling: Improved Resource-Augmentation Results," Proceedings of the ISCA International Conference on Computers and their Applications, Honolulu, Hawaii, March 2010. ,
(07 2010)

08/23/2011 16.00 Hennadiy Leontyev, S. Chakraborty, James Anderson. " Multiprocessor Extensions to Real-Time Calculus",
Real-Time Systems (08 2011)

08/23/2011 28.00 J. Anderson, J. Erickson. Beating G-EDF at its Own Game: New Results on G-EDF-likeSchedulers└,
IN submission (08 2011)

08/23/2011 29.00 C.Liu, J. Anderson. " Multiprocessor Schedulability Analysis for Self-Suspending Task Systems",
IN submission (08 2011)

08/23/2011 30.00 G. Elliott, C.H. Sun, J. Anderson. " Real-Time Handling of GPU Interrupts in LITMUSRT,
IN submission (08 2011)

TOTAL: 24

Number of Manuscripts:

Books

Received Paper

TOTAL:

Patents Submitted

Patents Awarded

Awards

Prof. James H. Anderson was elected ACM Fellow.

Prof. James H. Anderson was appointed W.R. Kenan Distinguished Professor of Computer Science.

Prof. James H. Anderson was elected IEEE Fellow.

Prof. Sanjoy Baruah was elected IEEE Fellow.

The paper "Making Shared Caches More Predictable on Multicore Platforms," by B. Ward, J. Herman, C. Kenna, and J. Anderson, received an Outstanding Paper Award at the 25th Euromicro Conference on Real-Time Systems in July 2013.

The paper "Fair Lateness Scheduling: Reducing Maximum Lateness in G-EDF-like Scheduling," by J. Erickson and J. Anderson, received the Best Paper Award at the 23rd Euromicro Conference on Real-Time Systems in July 2012.

The paper "Global Mixed-Criticality Scheduling on Multiprocessors," by H. Li and S. Baruah, received an Outstanding Paper Award at the 23rd Euromicro Conference on Real-Time Systems in July 2012.

The paper "The OMLP Family of Optimal Multiprocessor Real-Time Locking Protocols," by B. Brandenburg and J. Anderson, appeared in a special issue of Design Automation for Embedded Systems consisting of the best papers from the 2011 ACM International Conference on Embedded Software.

The paper "Globally Scheduled Real-Time Multiprocessor Systems with GPUs," by G. Elliott and J. Anderson, appeared in a special issue of Real-Time Systems consisting of the best papers from the 19th International Conference on Real-Time and Network Systems.

The paper "Optimality Results for Multiprocessor Real-Time Locking," by B. Brandenburg and J. Anderson, received the Best Student Paper Award at the 31st IEEE Real-Time Systems Symposium in December 2010.

The paper "Supporting Soft Real-Time Parallel Applications on Multiprocessors" was accepted for a special issue of the Journal of Systems Architecture on best papers from the 18th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications.

The paper "An Optimal k-Exclusion Real-Time Locking Protocol Motivated by Multi-GPU Systems," by G. Elliott and J. Anderson, appeared in a special issue of Real-Time Systems on best papers from the 19th International Conference on Real-Time and Network Systems.

The paper "Globally Scheduled Real-Time Multiprocessor Systems with GPUs," by G. Elliott and J. Anderson, appeared in a special issue of Real-Time Systems on best papers from the 18th International Conference on Real-Time and Network Systems.

Bjorn B. Brandenburg's Ph.D. dissertation received three awards: the 2012 Dean's Distinguished Dissertation Award in the area of Mathematics, Physical Sciences, and Engineering (a UNC award), the 2012 Council of Graduate Schools/ProQuest Distinguished Dissertation Award in Mathematics, Physical Sciences and Engineering (a national award), and the 2012 EDAA Outstanding Dissertation Award (an international award in embedded systems).

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Bjorn Brandenburg	0.03	
Glenn A. Elliott	0.16	
Jeremy Erickson	0.27	
Jonathan L. Herman	0.27	
Haohan Li	0.05	
Cong Liu	0.23	
Malcolm Mollison	0.15	
FTE Equivalent:	**1.16**	
Total Number:	**7**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
James H. Anderson	0.08	
Sanjoy Baruah	0.08	
FTE Equivalent:	**0.16**	
Total Number:	**2**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED	Discipline
Gary Bressler	0.00	Computer Science
FTE Equivalent:	**0.00**	
Total Number:	**1**	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME	
Jonathan L. Herman	
Christopher J. Kenna	
Malcom Mollison	
Total Number:	**3**

Names of personnel receiving PHDs

NAME	
Bjorn B. Brandenburg	
Haohan Li	
Cong Liu	
Total Number:	**3**

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

LITMUS-RT provides plugins for several real-time scheduling and synchronization algorithms. The scheduling algorithms that are provided can be used to host either hard or soft real-time workloads. In a hard real-time application, task deadlines cannot be missed. In a soft real-time application, some deadline tardiness is permissible. In the definition of "soft real-time" that we have focused on, deadline tardiness is required to be (provably) bounded. While the same scheduling algorithm can be used to schedule both types of applications (hard and soft), different analysis is required to prove real-time correctness for hard and soft real-time applications, as zero tardiness is required in the former case, but merely bounded tardiness in the latter. In this project, we identified the best scheduling and synchronization policies to use for both hard and soft real-time systems. The metric of comparison is schedulability: we seek algorithms that can correctly schedule the largest possible category of systems, with overheads considered. The phrase "correctly schedule" has a different interpretation for hard and soft real-time systems and it implies that analysis exists that can be used to predict which systems are schedulable. (That is, it should be possible to validate a system's timing constraints.)

The pre-existing LITMUS-RT implementation provided support for a variety of basic scheduling approaches, ranging from partitioning approaches (tasks are statically assigned to processors), to global approaches (tasks are scheduled from a single global runqueue), to clustered approaches (tasks are partitioned onto clusters of cores, with global scheduling used within each cluster). However, as its name suggests, LITMUS-RT was originally designed as a testbed that facilitated the empirical comparison of different scheduling and synchronization policies. Our goal in this project was to extend LITMUS-RT so that it is a fully-featured system that can host real applications. These extensions include: (i) new synchronization and scheduling algorithms that are better suited for large multicore platforms; (ii) new analysis and implementation methods for dealing with non-CPU resources such as graphics processing units (GPUs); (iii) support for dynamic task behavior (tasks can be created, destroyed, and their timing parameters altered at runtime); and (iv) mechanisms for supporting precedence constraints and task suspensions (e.g., due to I/0). Additionally, we have use (and are using) LITMUS-RT to experiment with real applications. Some efforts in this direction involving military avionics have already commenced in joint work with colleagues in industry and at research labs.

Significance

Multicore platforms provide significant processing power that could potentially be harnessed to support complex, resource-demanding real-time applications. However, for this to happen, appropriate real-time scheduling and synchronization algorithms must be devised, analyzed, and implemented within a working OS. That was precisely the objective of this project.

LITMUS-RT has matured to the point of being a very stable and useful platform. It has a growing international user base. It is now one of the standard de facto platforms for experimental research in the real-time systems research community. To date, it has been used by 20-30 research groups around the world.

Accomplishments

We have re-based LITMUS-RT to the latest Linux on (approximately) an annual bassis. Currently, it is based on Linux 3.10.5.

In work on mixed-criticality systems, we developed a scheduling framework for supporting tasks of different criticalities on multicore platforms. These are systems in which subsystems of differing criticalities exist: more critical subsystems require more conservative provisioning than less critical ones. For example, in avionics systems, flight control subsystems are more critical than mission control subsystems. For such systems, mixed-criticality scheduling algorithms have been proposed that attempt to reclaim system capacity lost to worst-case execution time pessimism. Such algorithms seem to hold great promise for multicore real-time systems, where such loss is particularly severe (due to difficulties in predicting contention with respect to shared hardware resources such as caches when assessing task execution times). Our group was actually first (as far as we know, only) group to propose and implement a mixed-criticality scheduling framework for multicore. As in other work on mixed-criticality scheduling (mostly uniprocessor work), our framework assumes that high-criticality tasks are provisioned conservatively and thus will generate significant processing "slack" at runtime. Our framework allocates this slack to low-criticality tasks so that their timing constraints can be met.

Using LITMUS-RT, we completed a major study on implementation tradeoffs affecting mixed-criticality systems using this framework. The unique nature of these mixed-criticality scheduling algorithms gives rise to a number of major challenges for the would-be implementer. In our study, we produced the first ever implementation (in LITMUS-RT) of a mixed-criticality scheduling framework on a multicore system. In this study, we experimentally evaluated design tradeoffs that arise when seeking to isolate tasks of different criticalities and to maintain overheads commensurate with a standard real-time OS. We also evaluated a key property needed for such a system to be practical: that the system be robust to breaches of the optimistic execution-time assumptions used in mixed-criticality analysis. This work was partially funded by AFOSR, but given its LITMUS-RT-oriented focus, some of the students involved were supported by this project.

Throughout the project, we have investigated research issues involving tardiness analysis for soft real-time systems scheduled using global real-time scheduling algorithms. We extended prior tardiness analysis so that task execution times can be specified stochastically and tardiness bounded in expectation. This result has important implications for soft-real-time applications. Such applications are typically provisioned assuming average-case, rather than worst-case, execution times. Prior analysis required that worst-case times be known. We also improved upon prior tardiness analysis by developing new analysis that allows lower tardiness bounds to be established than was previously possible. We also devised new tardiness analysis that can be applied in systems where tasks may suspend due to I/O operations, have precedence constraints (expressible using directed acyclic task graphs), and have non-preemptive sections.
We also extended our work on supporting graph-specified precedence constraints to be applicable to real-time workloads that are hosted across multiple nodes in a distributed system. We also devised new analysis that identifies a "best" global scheduling algorithm from the perspective of tardiness among a family of algorithms that from an implementation perspective are equivalent to the widely-studied global earliest-deadline-first (GEDF) algorithm. Surprisingly, we found that GEDF itself is not the best algorithm within this family. Additionally, we devised new tardiness analysis for task systems wherein tasks may be multi-threaded and for task systems in which consecutive invocations of a task may execute concurrently. A common theme in both cases is allowing intra-task parallelism. We also obtained new tardiness analysis for task systems wherein tasks may suspend execution to access external devices. Finally, we showed that by splitting jobs (i.e., task invocations) into sub-jobs, both observed tardiness and tardiness bounds can be reduced, even when the additional overheads of splitting jobs in practice are considered (overheads increase because tasks are preempted more frequently). This result follows because job splitting reduces task execution times and both tardiness bounds and observed tardiness are greater when execution times are greater.

Using LITMUS-RT, we completed a major experimental study that examines different real-time scheduling strategies on large multicore systems (in our study, a 24-core Intel platform was used). This work showed that, when real overheads are considered, partitioned schedulers are better to use for supporting hard real-time workloads, and clustered schedulers are better for soft real-time workloads (here, global scheduling is applied within clusters of four to eight cores).

We also completed a major study on the efficacy of "semi-partitioned" real-time scheduling algorithms. Such algorithms have been proposed as a compromise between pure global and partitioned approaches. Under a semi-partitioned algorithms, most tasks are assigned to processors (like under partitioning) but a few may migrate (like under global scheduling). This study evaluated the best (from an analytical perspective) semi-partitioned algorithms proposed to date and compared them, in terms of real-time schedulability with measured overheads considered, to pure partitioning and global approaches. Semi-partitioned scheduling was actually shown to be the preferred method for some workloads. However, semi-partitioned schedulers require an offline task-assignment phase, and this makes it difficult to apply such schedulers in systems where dynamic workload changes can occur.

We developed a number of real-time locking protocols that have provably optimal blocking behavior. These are the first such algorithms to be proven optimal. This work was mostly supported by NSF, but some of the students involved in this work were supported under this project. The protocols we have devised include spin-based and suspension-based locks for mutex synchronization and reader/writer synchronization, and extensions to these protocols that allow fine-grained lock nesting and replicated resources. We have conducted numerous LITMUS-RT experimental schedulability studies in which these various protocols were evaluated, with real overheads considered. (Summaries of many of these studies can be found in Bjorn Brandenburg's Ph.D. dissertation; a good overview of LITMUS-RT can be found there as well.)

Based on this synchronization work, we tackled the problem of making shared caches more predictable on multicore platforms. In safety critical domains, the usage of multicore platforms has been hampered by problems due to interactions across cores through shared hardware. The inability to precisely characterize such interactions can lead to pessimism in worst-case execution time analysis that is so great, the extra processing capacity of additional cores is entirely negated. Our work addresses this issue as it pertains to shared caches. Our approach utilizes an old idea called "page coloring," under which pages of physical memory are "colored" so that differently-colored pages cannot cause cache conflicts. We view such colors as shared resources: the real-time locking protocols just described are used to enable each task to lock its needed colors whenever it is invoked. We have implemented and evaluated this page-coloring approach using LITMUS-RT. We have since expanded our work on predictably managing shared non-CPU hardware resources like caches into a major NSF project.

In work that was partially funded by NSF, we produced an extension to LITMUS-RT called GPU-Sync, which is a framework for managing graphics processing units (GPUs) in multi-GPU multicore real-time systems. (GPU-Sync more than doubles the LITMUS-RT code base.) GPUSync was designed with flexibility, predictability, and parallelism in mind. Specifically, it can be applied under either static- or dynamic-priority CPU scheduling; can allocate CPUs/GPUs on a partitioned, clustered, or global basis; provides flexible mechanisms for allocating GPUs to tasks; enables task state to be migrated among different GPUs, with the potential of breaking such state into smaller "chunks"; provides migration cost predictors that determine when migrations can be effective; enables a single GPUâs different engines to be accessed in parallel; properly supports GPU-related interrupt and worker threads according to the sporadic task model, even when GPU drivers are closed-source; and provides budget policing to the extent possible, given that GPU access is non-preemptive. No prior real-time GPU management framework provides a comparable range of features. GPU-Sync utilizes the aforementioned optimal real-time locking protocols to arbitrate access to GPU-related resources.

Conclusions

In our algorithmic research, we tackled fundamental resource-allocation problems that need to be addressed to enable both soft and hard real-time workloads to be hosted on multicore platforms. In our OS and prototyping efforts, we showed that such algorithms are not merely of theoretical interest, but can be practically applied.

Technology Transfer:

Our work on mixed-criticality systems was done jointly with colleagues at Northrop-Grumman Corp. Also, we did some joint work on data-stream processing on multicore platforms, where soft real-time constraints apply, with a colleague at AT&T Research. We also did some joint work with colleagues at a company called Real-Time Innovations on real-time computing on "many-core" (10s to 1000s of cores on a chip) platforms that resulted in Phase 1 STTR funding from the U.S. Air Force Research Laboratory. Our GPU-related work has recently resulted in new funding from General Motors Corp.

Technology Transfer