Real-Time Dwell Scheduling of Component-Oriented Phased Array Radars*

Tei-Wei Kuo, Yung-Sheng Chao[†], Chin-Fu Kuo, Cheng Chang[†], and Yu-Lin Su[†]

Department of Computer Science and Information Engineering

National Taiwan University, Taipei, Taiwan 106, ROC

[†]Electronic System Division

Chung Shan Institute of Science and Technology, Lung-Tan, Taiwan 325, ROC

Abstract

A multi-function phased array radar must search and track suspicious targets in its surveillance space in a real-time fashion. This paper targets one of the most important issues in the design of modern phased array radars: real-time dwell scheduling. We propose a rate-based approach to schedule radar dwells in a real-time fashion. We show how to reserve radar resources to guarantee the minimum radar operation without sacrificing the stability of the system. The strength of our approach is verified by a series of simulation experiments based a real phased array radar for air defense frigates [8].

1 Introduction

A multi-function phased array radar must search and track suspicious targets in its surveillance space in a real-time fashion. There are two major modules in a phased array radar: Radar Control Computer (RCC) and Signal Processor (SP). RCC is responsible to scheduling radar beam transmissions for searching and tracking based on targets' status and search types. SP must process returned signals in a real-time fashion. With the advance of software and hardware technology, a modern phased array radar is no longer a complicated hardware system with everything wired. Instead, engineers are now building phased array radars with commercial-off-the-shelf (COTS) components, and the functions of many hardware components are now re-implemented by software modules [5].

Many existing (or even advanced) phased array radar systems still adopt inefficient non-real-time re-

*Supported in part by research grants from the Chung-Shan Institute of Science and Technology and National Science Council under Grants NSC89-2213-E-002-200

Figure 1: The hardware architecture of a component-oriented phased array radar

source scheduling mechanisms, such as those based on FIFO-like, simple priority-driven, or cyclic-executive-like scheduling algorithms [2]. The common reason in doing so is mainly because of hardware constraints. As a result, much resource is wasted with a very limited guarantee on system performance. Although researchers and engineers have started exploring real-time dwell scheduling at RCC, the proposed algorithms are mainly variations of the Partial Template algorithm. The major drawback of existing work on the Partial Template algorithm is on their restriction in resource utilization. Various inefficient scheduling mechanisms or data structures such as FIFO queues were adopted such that the deadlines of critical tasks are difficult to guarantee, due to lengthy priority inversion time, where priority inversion is a phenomenon in which a higher-priority task is blocked by a lower-priority task because of resource synchronization. The situation is further worsened by the highly dynamic nature of radar workloads. In order to guarantee the

minimum radar operation and to maximize the radar performance, engineers usually use a large number of radar resources (including computing power) which may not be needed under proper real-time scheduling algorithms to accomplish their performance goals. This observation underlines the motivation of this research. The system architecture under investigation is as shown in Figure 1.

Rate-based scheduling has been an active research topic in the past few years for real-time resource allocation. Researchers proposed various rate-based scheduling algorithms for periodic and sporadic processes based on the notion of General Processor Sharing (GPS) [1, 12, 17, 24, 28, 29, 30]. The idea of GPS-based scheduling is very different from common disciplines such as priority-driven scheduling [3, 19, 20, 27] and time-driven scheduling [9]. The GPS-based scheduling is a work-conserving scheduling mechanism. The schedulability of each process in GPS-based systems is guaranteed with an assigned CPU service rate, independent of the demands of other processes. The enforcement of a guaranteed CPU service rate for a process must rely on certain admission control mechanism to manage the total workload of the system [25].

This paper targets an essential issue for the design of modern phase array radars, i.e., real-time dwell scheduling. Two important issues in real-time dwell scheduling are addressed: (1) the minimum operation guarantee of phase array radars, and (2) the system capacity estimation. We propose a rate-based approach to guarantee the minimum operation of a phased array radar and show that our approach can effectively schedule radar tasks with a good radar resource utilization. Our methodology is verified by a series of experiments based on a phased array radar for air defense frigates [8], for which we have very encouraging results. We show that real-time technology does improve radar performance significantly. In particular, our proposed approach can greatly improve the performance of a phased array radar by tracking over 116% more targets than the traditional dwell scheduling approaches and, at the same time, improve missile tracking by over 40%, compared to that under traditional ones.

The rest of this paper is as follows: Section 2 presents our rate-based approach in real-time dwell scheduling and our methodology in resource reservation. In Section 3, a series of simulation experiments were done based on a real example system. Section 4 is the conclusion.

2 Real-Time Dwell Scheduling

2.1 Objectives

In a typical phased array radar, RCC schedules tasks in a real-time fashion by sending SP radar beam commands; SP then controls the electronic Beam Steering Controller (BSC) and the Transmitter to transmit microwave through Antenna. RCC schedules tasks in units called *scheduling interval* (SI) [2, 7, 10]. RCC sends a sequence of commands to SP for dwell transmissions in the beginning of a SI and retrieves results from SP as well, as shown in Figure 1.

In this paper, we propose a rate-based approach in the guarantee of the minimum system operation and demonstrate the capability of real-time scheduling technology, especially on rate-based scheduling, in improving the performance of a modern phased array radar. Two important issues in real-time dwell scheduling are addressed: (1) the minimum operation guarantee of phase array radars, and (2) the system capacity estimation.

2.2 Radar-Resource Reservation Methodology

We first define a reservation ratio θ_i for each task under the minimum operation requirements. In the next section, we shall propose an efficient scheduling algorithm called *Batch-TC* for dwell scheduling at RCC. The minimum operation requirements of a phased array radar might consist of a collection of High-Priority and Low-Priority Search tasks and a minimum number of targets which the system must be able to track simultaneously.

Let $\Pi = \{\tau_{HS}^1, \tau_{HS}^2, \dots, \tau_{HS}^N\}$ be a collection of High-Priority Search tasks necessary for the minimum operation of the system (e.g., for Horizon Search or Long Range Search). Note that Low-Priority Search tasks are often done by a best effort. The reservation ratio θ_{HS}^i of each High-Priority Search task τ_{HS}^i is set as

$$\frac{B_{HS}^i C_{HS}^i}{P_{HS}^i},$$

where B_{HS}^i , C_{HS}^i , and P_{HS}^i are the number of beams, dwell length, and period of τ_{HS}^i , respectively. In each j th period P_{HS}^i of τ_{HS}^i , τ_{HS}^i initiates B_{HS}^i instances, where each instance (i.e., a beam transmission and processing) has a dwell length C_{HS}^i . In each j th period P_{LS}^i of a Low-Priority Search task τ_{LS}^i , τ_{LS}^i initiates B_{LS}^i instances, where each instance has a dwell length C_{LS}^i . Low-Priority Search tasks do not have a reserva-

tion ratio because they only utilize unused dwell time in each SI.

Because the tracking of an identified target starts with a Confirmation Track, a sequence of Normal Track dwells, and possibly a sequence of Precision Track dwells, the reservation ratio of each tracked target is defined as

$$\max\left\{\frac{C_{TC}}{D_{TC}}, \frac{C_{NT}}{(P_{NT}^L - DT_{NT})}, \frac{C_{PT}}{(P_{PT}^L - DT_{PT})}\right\},$$

where C_{TC} and D_{TC} are the dwell length and relative deadline of a Track Confirmation, respectively, and C_{NT} , DT_{NT} , and P_{NT}^L (C_{PT} , DT_{PT} , and P_{PT}^L) are the dwell length, the dormant time, and the lower bound of the period of a Normal Track (/Precision Track), respectively. The dormant time is to leave enough slack for the RCC scheduler to schedule dwells in the next SI. In fact, Normal and (High) Precision Track dwells are all *semi-periodic* tasks, where *semi-periodic* means "periodic" but with dynamically changing periods. The reason for tracking being semi-periodic is because the distance between every two consecutive tracking executions for a target depends on many factors, such as target type, target position, target speed, etc. Note that after a number of Normal Track instances, the track task τ_T^i might either stop (because no tracking is needed any more) or become a Precision Track task (because a higher tracking precision is needed).

A High Precision Track task, such as that for missile guidance, is also semi-periodic and issues one beam per period. The reservation ratio of a High Precision Track task is defined as

$$\frac{C_{HPT}}{(P_{HPT}^L - DT_{HPT})},$$

where C_{HPT} , DT_{HPT} , and P_{HPT}^L are the dwell length, the dormant time, and the lower bound of the period of a High Precision Track, respectively. The above settings of reservation ratios consider the worst-case resource reservation.

2.3 Real-Time Dwell Scheduling Methodology - the Batch-TC Algorithm

Consider a task τ_i with a reservation ratio θ_i . Each job of the task is given a *virtual deadline* for EDF scheduling, where EDF is a well-known real-time scheduling algorithm, and EDF schedules the job with the most urgent deadline first [19]. The Batch-TC algorithm is a batch version of the TB-server (and CUS) scheduling algorithm [25], where dwell scheduling is done every SI, and no preemption is allowed:

Suppose that the j_{th} job of τ_i arrives at time t , and the virtual deadline of the $(j-1)_{th}$ job of τ_i is $d_{i,j-1}$. The virtual deadline $d_{i,j}$ of the j_{th} job of τ_i is defined as follows: Suppose that the dwell length of the job is $c_{i,j}$.

$$d_{i,j} = \max\{t, d_{i,j-1}\} + \frac{c_{i,j}}{\theta_i},$$

By definition, $d_{i,0} = 0$. The request is inserted into the ready queue of the system and scheduled by EDF in the next SI. When there are multiple instances of a task (such as search tasks) arriving at the same time, their deadlines are set as above and all inserted into the ready queue. Note that only dwells which can fit in one SI are scheduled in the next SI, and the scheduling of the rest will be delayed until the end of the next SI.

Example 1 A Batch-TC Schedule

Suppose that one High-Priority Search task, two High-Precision Track tasks, two Precision Track tasks, and one Normal Track task arrive at the beginning of the i_{th} SI, as shown in Figure 2. Let the High-Priority Search task τ_{HS} issue 45 instances (For the simplicity in the presentation of the Batch-TC algorithm, assume that all instances of τ_{HS} issued in previous SI's were processed).

The timing constraints of all tasks are as shown in Table 1. According to the definitions of the reservation ratios, the reservation ratio of τ_{HS} is $(B_{HS}^i C_{HS}^i) / P_{HS}^i = (45 * 6ms) / 1000ms = 0.27$. Let the dormant time of every track task be one SI, i.e., 25ms. The reservation ratio for each High-Precision Track task is derived by the following formula:

$$\frac{C_{HPT}}{(P_{HPT}^L - DT_{HPT})} = \frac{2ms}{75ms} = 0.02666$$

Let the dwell length C_{TC} and deadline D_{TC} of a Track Confirmation be 6ms and 20SI, respectively. The reservation ratio for each tracked target is derived by the following formula:

$$\max\left\{\frac{C_{TC}}{D_{TC}}, \frac{C_{NT}}{(P_{NT}^L - DT_{NT})}, \frac{C_{PT}}{(P_{PT}^L - DT_{PT})}\right\} = 0.05333$$

The virtual deadlines of the first four instances of τ_{HS} , the two High-Precision Track tasks τ_{HPT}^1 and τ_{HPT}^2 , the Precision Track task τ_{PT}^1 , and the two Normal Track tasks τ_{NT}^1 and τ_{NT}^2 are listed in Figure 2.

The tasks are sorted in the ready queue of the Batch-TC scheduling algorithm and scheduled in the i_{th} SI, as shown in Figure 2. Note that only τ_{HS}^1 , τ_{HS}^2 , τ_{HS}^3 , τ_{PT}^1 , τ_{HPT}^1 , and τ_{HS}^4 are scheduled in the i_{th} SI. Most importantly, the execution of τ_{HS}^4 crosses the i_{th} and $(i+1)_{th}$ SI's. In other words, because of the non-preemptibility

Tasks	Number of Dwells	Dwell Length	Period	Reservation Ratio
High-Priority Search task (τ_{HS})	45	6ms	40SI	0.27
High-Precision Track tasks (τ_{HPT}^1 and τ_{HPT}^2)	1	2ms	(4SI, 10SI)	0.02666
Normal/Precision Track tasks (τ_{PT}^1 , τ_{PT}^2 , and τ_{NT}^1)	1	$C_{PT} = 4ms$ $C_{NT} = 4ms$	(4SI, 10SI) for τ_{PT}^1 and τ_{PT}^2 (10SI, 80SI) for τ_{NT}^1	0.05333

Table 1: Task parameters (1SI = 25ms).

Tasks	Virtual Deadline
τ_{HS}^1	$SI_i + 22.22$
τ_{HS}^2	$SI_i + 44.44$
τ_{HS}^3	$SI_i + 66.66$
τ_{HS}^4	$SI_i + 88.88$
τ_{HPT}^1	$SI_i + 75.018$
τ_{HPT}^2	$SI_i + 75.018$
τ_{PT}^1	$SI_i + 75.005$
τ_{NT}^1	$SI_i + 75.005$
τ_{NT}^2	$SI_i + 75.005$

Table 2: The virtual deadlines of tasks ready at the beginning of SI_i (1SI = 25ms).

Figure 2: A Batch-TC schedule

of dwells, RCC can not schedule any urgent dwell in the beginning of the $(i+1)_{th}$ SI until τ_{HS}^4 finishes. Such a blocking phenomenon can be modeled by the blocking due to the global resource synchronization. $\square$

We can show that, given a set of radar tasks under the minimum operation requirements, if their total reservation ratio is no larger than $(1 - \max_{j \geq 1} \{B_j / \delta_j\})$, then every radar task can issue the specified number of beams before its deadline, where B_j is the maximum dwell length of all radar tasks, and δ_j is the shortest relative deadline of all dwells.

Note that the reservation ratio for each tracking task considers the worst-case radar resource utilization. Due to different resource requirements of tracking tasks at different stages, a single reservation ratio seems restrictive in the entire system utilization. The method-

ology proposed in this paper can be easily extended to let several tasks sharing a reservation ratio to maximize the system utilization and to guarantee a better system capacity. Furthermore, because the reservation ratio for each radar task is based on its worst-case resource utilization, i.e., the maximum usage of the radar resource, we can also extend the the proposed mechanism to reclaim the wasted radar resource, whenever possible, while guaranteeing the minimum operation of a radar.

3 Performance Evaluation

3.1 System Model and Data Sets

Our methodology is verified by a series of experiments based on a phased array radar for air defense frigates [8], for which we have very encouraging results. The parameters of tasks are summarized in Table 3. The High-Priority Search task was for Horizon Search. Track Confirmation tasks, which were not periodic, had a deadline equal to 20SI. There was one Low-Priority Search task (for Normal Volume Search) in the system,

Task Types	Dwell Length	Beams per Period	Period	Priority
High-Priority Search	6ms	45 beams	40SI	1
Track Confirmation	6ms	1 beam	Deadline = 20SI	2
High-Precision Track	2ms	1 beam	(4SI, 10SI)	3
Precision Track	4ms	1 beam	(4SI, 10SI)	4
Normal Track	4ms	1 beam	(10SI, 80SI)	5
Low-Priority Search	2ms	20 beams	40SI	6

Table 3: Task parameters of simulation experiments (1SI = 25ms).

which issued 20 beams for every 40SI, and each dwell length was 2ms. In the experiments, the ratio of track tasks and High-Precision Track tasks remained 2 : 1.

The primary performance metric is the ratio of requests that miss deadlines, referred to as the *Miss Ratio*. Let num_i and $miss_i$ be the total number of task requests and deadline violations during an experiment, respectively. *Miss Ratio* is calculated as $\frac{miss_i}{num_i}$.

The experiments described are meant to assess the capability of the proposed algorithm Batch-TC with reservation ratio reclamation in dwell scheduling. We have implemented a simulation model for a multi-function phased array radar. We compare the performance of the Batch-TC algorithm (with reservation ratio reclamation), the earliest deadline first algorithm (EDF), and the traditional Partial Template scheduling algorithm (PM), where the problem of EDF is on the handling of transient overload.

3.2 Experimental Results

Figure 3: The miss ratio of the High-Priority Search task.

Figure 3 shows the miss ratio of the High-Priority Search task. Since Batch-TC and PM both guaranteed its service, its miss ratio remained zero, regardless of the number of tracking tasks in the system. On the other hand, EDF started missing most of the deadlines of the High-Priority Search task when the number of tracking tasks in the system was over 20.

Figure 4: The miss ratio of High-Precision Track tasks.

Figure 4 shows the miss ratio of High-Precision Track tasks. Since the ratio of track tasks and High-Precision Track tasks remained 2 : 1, Batch-TC and EDF could both schedule up to 7 High-Precision Track tasks without missing any of their deadlines, and the miss ratio of High-Precision Track tasks remained low until the number of High-Precision Track tasks was around 10, i.e., the number of tracking tasks was over 30. On the other hand, PM could only schedule 4 High-Precision Track tasks without missing any of their deadlines (here High-Precision Track tasks were favored under PM).

Figure 5 shows the miss ratio of all track tasks. Batch-TC and EDF also greatly outperformed PM.

Figure 5: The miss ratio of all track tasks (including Normal Track, Track Confirmation, Precision Track tasks), without an even search workload distribution.

Figure 6 shows the miss ratio of the Low-Priority Search task. It was surprised to see that Batch-TC always met the deadlines of the Low-Priority Search task, regardless of the number of tracking tasks. It was because the period of the Low-Priority Search task was long, and Batch-TC happened to have bandwidth left in the reservation for tracking tasks and the High-Priority Search task.

We show that real-time technology does improve radar performance significantly. In particular, our proposed approach can greatly improve the performance of a phased array radar by tracking over 116% more targets than the traditional dwell scheduling approaches such as the Partial Template algorithm and, at the same time, improve missile tracking by over 40%, compared to that under traditional ones, as shown in the above figures.

4 Conclusion

Past work in real-time dwell scheduling suffers from serious resource utilization problems mainly because of hardware constraints. Much radar resource is wasted without significant performance improvement. This paper targets an essential issue for the design of modern phase array radar, i.e., real-time dwell scheduling. We propose a rate-based real-time dwell scheduling methodology and show its effectiveness in guaranteeing the timing constraints of radar workloads. The proposed approach is verified by a series of experiments based on a real phased array radar for air defense

Figure 6: The miss ratio of the Low-Priority Search task, without an even search workload distribution.

frigates [8], for which we have very encouraging results. We show that real-time technology does improve radar performance significantly. In particular, under the tested case, the proposed Batch-TC could greatly improve the performance of a phased array radar by tracking over 116% more targets than the traditional approach and, at the same time, improving the capacity of High-Precision tracking by over 40%, compared to that under PM.

We believe that with the complexity of advanced radar systems people are building, only rigorous theory which lends itself to advanced implementation methods can provide us with the reliability and performance acceptable to users. For the future work, we will explore real-time dwell scheduling of a multi-SP radar system and tune up our methodology for radar systems of different scales and goals.

Acknowledgements: We would like to thank ear Mr. S.C. Wan and Dr. Tai-Chung Wang for their help in this work.

References

- [1] L. Abeni and G. Buttazzo, "Integrating Multimedia Applications in Hard Real-time Systems," *IEEE 19th Real-Time Systems Symposium*, December 2-4, 1998.
- [2] R.A. Baugh, "Computer Control of Modern Radars," RCA M&SR-Moorestown Library, 1973.
- [3] T.P. Baker, "A Stack-Based Resource Allocation Policy for Real Time Processes," *IEEE 11th Real-Time Systems Symposium*, December 4-7, 1990.

- [4] A. Barbato and P. Giustiniani, "An Improved Scheduling Algorithm for a Naval Phased Array Radar," *ALENIA Defence Systems*, Italy.
- [5] C. Chang and T.-C. Wang, "Use Object-Oriented Paradigm to Design a Programmable Radar Digital Signal Processor," *Third Workshop on Object-Oriented Technology and Applications (OOTA' 97)*, September 1997.
- [6] Z. Deng and J. W.-S. Liu, "Scheduling Real-Time Applications in an Open Environment," *IEEE 18th Real-Time Systems Symposium*, December 1997.
- [7] R. Filippi and S. Pardini, "An Example of Resources Management in A Multifunctional Rotating Phased Array Radar," *Real-Time Management of Adaptive Radar Systems, IEE Colloquium*, 1990, pp. 2/1-2/3.
- [8] A.G. Huizing and A.A.F. Bloemen, "An Efficient Scheduling Algorithm for a Multifunction Radar," *IEEE International Radar Conference*, 1996, pp. 359-364.
- [9] C.-W. Hsueh and K.-J. Lin, "On-Line Schedulers for Pinwheel Tasks Using the Time-Driven Approach," *the 10th Euromicro on Real-Time Systems*, June 1998, pp. 180-187.
- [10] A. Izquierdo-Fuente and J.R. Casar-Corredera, "Approach to Multifunction Radar Scheduling Simulation," *IEEE Telesystems Conference*, 1994, pp. 67-70.
- [11] A. Izquierdo-Fuente and J.R. Casar-Corredera, "Optimal Radar Pulse Scheduling using a neural Network," *IEEE International Conference on Volume:7*, 1994, pp. 4588-4591.
- [12] K. Jeffay and S. Goddard, "A Theory of Rate-Based Execution," *IEEE 20th Real-Time Systems Symposium*, Dec 1999, pp. 304-314.
- [13] G. V. Keuk and S. S. Blackman, "On Phased-Array Radar Tracking and Parameter Control," *IEEE Transactions*, Jan 1993, pp. 186-194.
- [14] K. Jeffay and S. Goddard, "A Theory of Rate-Based Execution," *IEEE 20th Real-Time Systems Symposium*, December 1999.
- [15] T.-W. Kuo and C.-H. Li, "A Fixed-Priority-Driven Open Environment for Real-Time Applications," *the IEEE 20th Real-Time Systems Symposium*, Phoenix, USA, 1999.
- [16] D.-I. Kang, R. Gerber, and M. Saksena, "Performance-Based Design of Distributed Real-Time Systems," *IEEE 1997 Real-Time Technology and Applications Symposium*, June 1997, pp. 2-13.
- [17] T.-W. Kuo, W.-R. Yang, and K.-J. Lin, "EGPS: A Class of Real-Time Scheduling Algorithms Based on Processor Sharing," *Proceeding of the 10th Euromicro Workshop on Real-Time Systems*, pp. 27-34, June 1998.
- [18] T.-W. Kuo, C.-F. Kuo, and C. Chang, "Real-Time Digital Signal Processing of Component-Oriented Phased Array Radars," *IEEE 21st Real-Time Systems Symposium*, Orlando, FL, USA, 2000.
- [19] C.L. Liu and J.W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *JACM*, Vol. 20, No. 1, January 1973, pp. 46-61.
- [20] A.K. Mok, "Fundamental Design Problems for the Hard Real-Time Environment," MIT Ph.D. Dissertation, Cambridge, MA, 1983.
- [21] R.L. Nevin and F.W. Schatz, "AN/APG-67 Multi-mode Radar Development," *IEEE International Radar Conference*, 1985, pp. 1-8.
- [22] A.K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case," *Proceedings of IEEE INFOCOM*, 1992.
- [23] "Rapid prototyping of Application Specific Signal Processors" (RASSP), <http://eto.sysplan.com/ETO/RASSP>.
- [24] M. Spuri and G. Buttazzo, "Scheduling Aperiodic Tasks in Dynamic Priority Systems," *IEEE 17th Real-Time Systems Symposium*, pp. 179-210, December 1996.
- [25] M. Spuri, G. Buttazzo, and Sensini, "Scheduling Aperiodic Tasks in Dynamic Scheduling Environment," *IEEE Real-Time Systems Symposium*, 1995.
- [26] D. Stromberg and P. Grahm, "Scheduling of Tasks in Phased Array Radar," *IEEE International Radar Conference*, 1996, pp. 318-321.
- [27] L. Sha, R. Rajkumar, and J.P. Lehoczky, "Priority Inheritance Protocols: An Approach to Real-Time Synchronization," *Technical Report CMU-CS-87-181*, Dept. of Computer Science, CMU, November 1987. *IEEE Transactions on Computers*, Vol. 39, No. 9, September 1990.
- [28] I. Stoica, H. Abdel-Wahab, K. Jeffay, S.K. Baruah, J.E. Gehrke, and C.G. Plaxton, "A Proportional Share Resource Allocation Algorithm for Real-Time, Time-Shared Systems," *IEEE Real-Time Systems Symposium*, 1996, pp. 288-299.
- [29] C.A. Waldspurger and W.E. Weihl, "Stride Scheduling Deterministic Proportional Share Resource Management," *Technical Memorandum*, MIT/LCS/TM-528, Laboratory for CS, MIT, July 1995.
- [30] C.A. Waldspurger, "Lottery and Stride Scheduling: Flexible Proportional-Share Resource Management," Ph.D. Thesis, Technical Report, MIT/LCS/TR-667, Laboratory for CS, MIT, September 1995.
- [31] T.-C. Wang, "A Programmable Parallel Radar Signal Processor - PRSP", *Proceedings of 1996 Workshop on Distributed System Technologies and Applications*, May 1996, pp 125-129.