CPU Scheduling for Active Processing using Feedback Deficit Round Robin

Tilman Wolf and Dan Decasper
{wolf, dan}@arl.wustl.edu
Applied Research Laboratory
Washington University
St. Louis, Missouri 63130

1 Introduction

To address the increasing demand for customized and application-specific processing on network routers, "active networks" have been proposed [1]. Active network routers are capable of processing code, which is either carried as part of the datagram's payload or referenced by a specific value in the datagram. Execution Environments (EEs) provide the context in which active processing is performed.

Process and thread scheduling for Execution Environments differs from regular time-slice scheduling as implemented by modern general-purpose operating systems. Active packets can contain individual code, which leads to context switches (additional to the context switches caused by scheduling) when packets with different code are processed in EEs. Contrary to general-purpose OSs, where this happens only when new applications are launched, these additional context switches can occur frequently when many packets contain individual code.

Context switching times of several thousand cycles are typical for switching between different processes [2]. This represents a considerable overhead in active networks, where CPU resources are already scarce. To reduce unnecessary context-switches, we propose Feedback Deficit Round Robin (FDRR) for CPU scheduling, which stems from Deficit Round Robin (DRR) [3], a fair-queuing scheme for packet scheduling.

2 Feedback Deficit Round Robin

FDRR uses the individual *expected processing time* as a criteria for CPU cycle management similar to the packet length used by DRR for bandwidth management. Since the individual processing time cannot be known ahead of time, an estimate is used. A feedback mechanism applies the measured processing time to adjust the estimate.

The outline of FDRR is shown in Figure 1. Each Execution Environment has a queue where packets are stored for processing. For each queue, a deficit counter and an estimate is maintained. The deficit represents the amount of processing that a particular queue can use. The estimate represents the expected processing time for the next packet.

The DRR scheduler forwards packets of a queue to the processor as long as the deficit is larger than the estimate of the next packet. With each packet, a timer is started that interrupts the processor in case a packet exceeds its resources. When the processing is finished or terminated, the actual processing time is used to adjust the deficit, as well as the estimate that is used for the next packet.

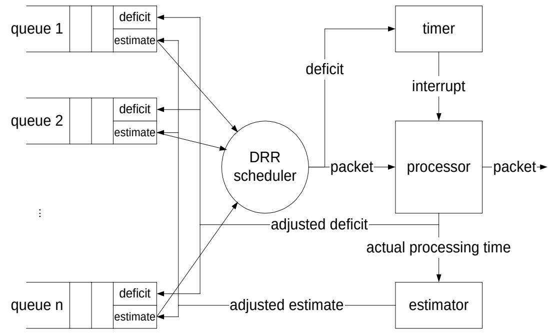

Figure 1: Feedback Deficit Round Robin

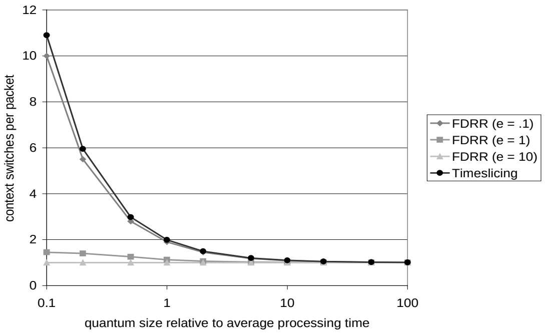

Figure 2: Number of Context Switches

3 Simulation Results

We assume for our measurements that each active packet can carry individual code. This means that a 'context switch' happens every time the processing of a new packet is started, because the instruction cache will not have the instructions cached, and state information from earlier packets has to be made available.

To compare the performance of FDRR with the timesliced scheduling, we simulated both algorithms over a range of parameters. The number of context switches per packet are shown in Figure 2. The quantum / timeslice ranges from .1 times to 100 times the average processing time of a packet. FDRR is simulated using three different estimates (.1, 1, and 10 times the average actual processing time).

Timeslicing causes most context switches for any quantum size. If FDRR underestimates the processing ($e = .1$) significantly, it performs similar to Timeslicing. If the estimation is the actual processing time ($e = 1$), FDRR incurs significantly fewer context switches for quantum sizes in the range of the actual processing time. FDRR causes the least number of context switches for large overestimations ($e = 10$). In this case, the packet is practically always processed to completion, and no interrupts or context switches due to scheduling occur. This case comes closest to the ideal number of context switches per packet of 1. For larger quantum sizes, many packets are being processed in the same round, and scheduling differences between FDRR and Timeslicing disappear.

Although FDRR with $e = 10$ seems to perform best, it also incurs most delay. While in Timeslicing a packet is processed immediately when it reaches the head of the queue, in FDRR the deficit is accumulated until it reaches the estimated processing time. This delay is significant for large overestimations. Thus, FDRR with a good estimation ($e = 1$) has the best overall performance.

References

[1] Tennenhouse, D., Smith, J., Sincoskie, D., Wetherall, D. [1997]. "A Survey of Active Network Research," *IEEE Communications*, 35, 1, January 1997, 80-86.

[2] Padmanabhan, V., Roselli, D. [1994]. "The Real Cost of Context Switching," *http://http.cs.berkeley.edu/~padmanab/class_projects/cs252.ps.*

[3] Shreedhar, M., Varghese, G. [1995]. "Efficient Fair Queuing using Deficit Round Robin," *Proc. of SIGCOMM 95*, ACM, Cambridge, Mass.