Weighted Differential Scheduler

Hans Eberle, Wladek Olesinski
Oracle Labs

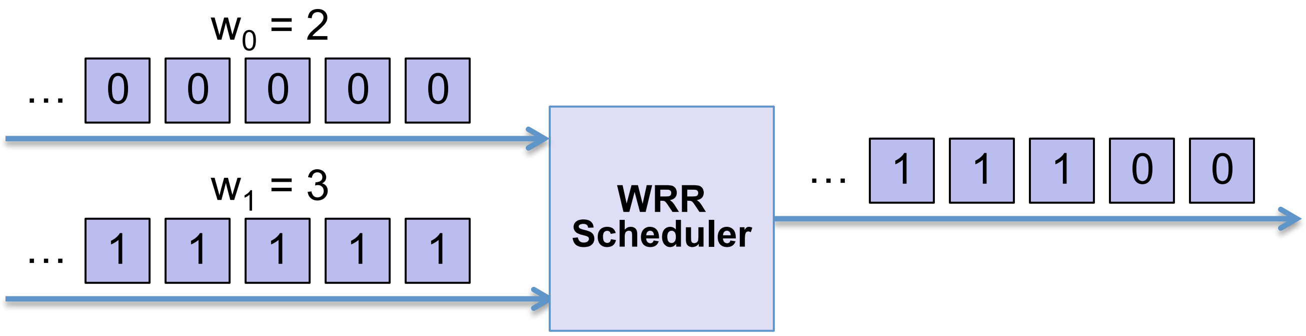
Outline

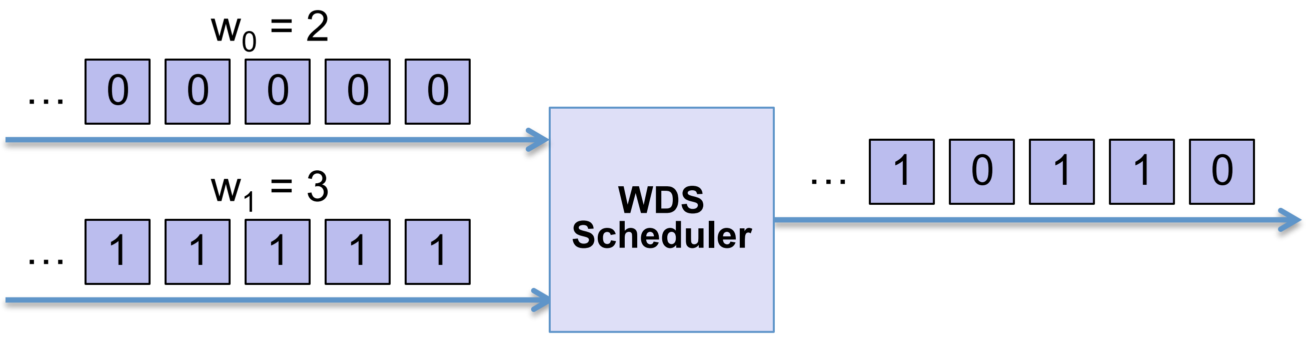
- Motivation
- Digital Differential Analyzer
- Weighted Differential Scheduler (WDS)
- Simulation Results
- Summary

Motivation

- Problem statement
  - Schedule use of a shared resource among several requesters
  - Weights determine requesters' share of the resource
  - Access to the shared resource in fixed-size time intervals
- Goal
  - Requester's intervals are spaced apart at equal distances
  - Reduce forwarding latency and minimize buffer memory capacity
- Applications
  - Packet forwarding in network switches
  - Process scheduling etc.

Jitter

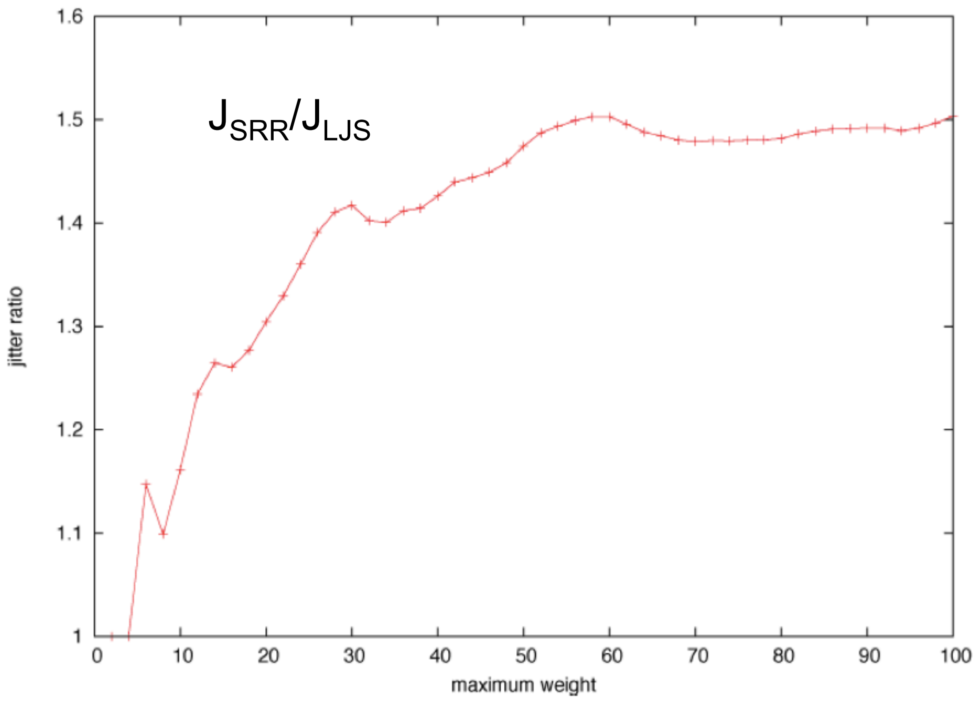

$w_0 = 2$
… 0 0 0 0 0
$w_1 = 3$
… 1 1 1 1 1
WRR Scheduler
… 1 1 1 0 0


$w_0 = 2$
… 0 0 0 0 0
$w_1 = 3$
… 1 1 1 1 1
WDS Scheduler
… 1 0 1 1 0

Related Work

Smoothed Round Robin (SRR)

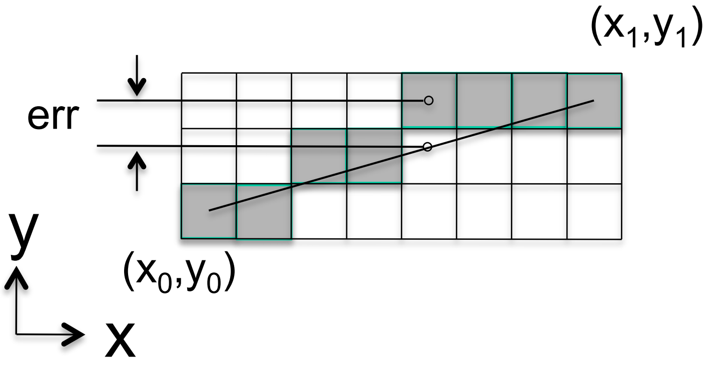
Low-Jitter Scheduler (LJS)



Digital Differential Analyzer

- Used in computer graphics for rasterizing lines, e.g. for interpolating the raster coordinates of the line given a start and end point
- Example: Bresenham line drawing algorithm
- Avoids drift, i.e. accumulation of rounding errors
- Can be implemented with integer additions and subtractions

Bresenham Line Drawing Algorithm



```
function DrawLine $((x_0,y_0),(x_1,y_1))$
    $x \Leftarrow x_0$
    $y \Leftarrow y_0$
    $err \Leftarrow 0$
    $deltaerr \Leftarrow (y_1 - y_0)/(x_1 - x_0)$
    while $x < x_1$ do
        plot(x,y)
        $x \Leftarrow x + 1$
        $err \Leftarrow err + deltaerr$
        while $err \geq 0.5$ do
            $y \Leftarrow y + 1$
            $err \Leftarrow err - 1$
        end while
    end while
```

Scheduling Algorithm for Two Flows

WDS

```
$err \Leftarrow 0$
$deltaerr \Leftarrow w_1/w_0$
loop
    serve $f_0$
    $err \Leftarrow err + deltaerr$
    while $err \geq 0.5$ do
        serve $f_1$
        $err \Leftarrow err - 1$
    end while
end loop
```

Bresenham

```
$err \Leftarrow 0$
$deltaerr \Leftarrow (y_1 - y_0)/(x_1 - x_0)$
while $x < x_1$ do
    plot(x,y); $x \Leftarrow x + 1$
    $err \Leftarrow err + deltaerr$
    while $err \geq 0.5$ do
        $y \Leftarrow y + 1$
        $err \Leftarrow err - 1$
    end while
end while
```

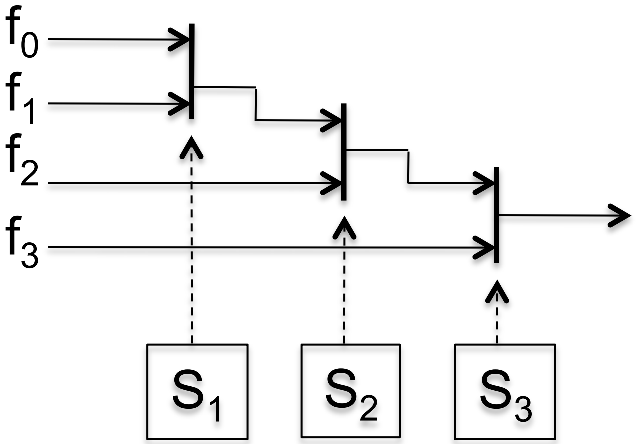
Using Integer Arithmetic Only

```
err1 ⇐ 0
deltaerr1 ⇐ w1/w0          | *2w0
loop
    serve f0
    err1 ⇐ err1 + deltaerr1
    while err1 ≥ 0.5 do    | *2w0
        serve f1
        err1 ⇐ err1 – 1    | *2w0
    end while
end loop
```

```
err1 ⇐ 0
deltaerr1 ⇐ 2*w1
loop
    serve f0
    err1 ⇐ err1 + deltaerr1
    while err1 ≥ w0 do
        serve f1
        err1 ⇐ err1 – 2*w0
    end while
end loop
```

Scheduling > 2 Flows – Iterative Method

- Iteratively merge two flows at the time

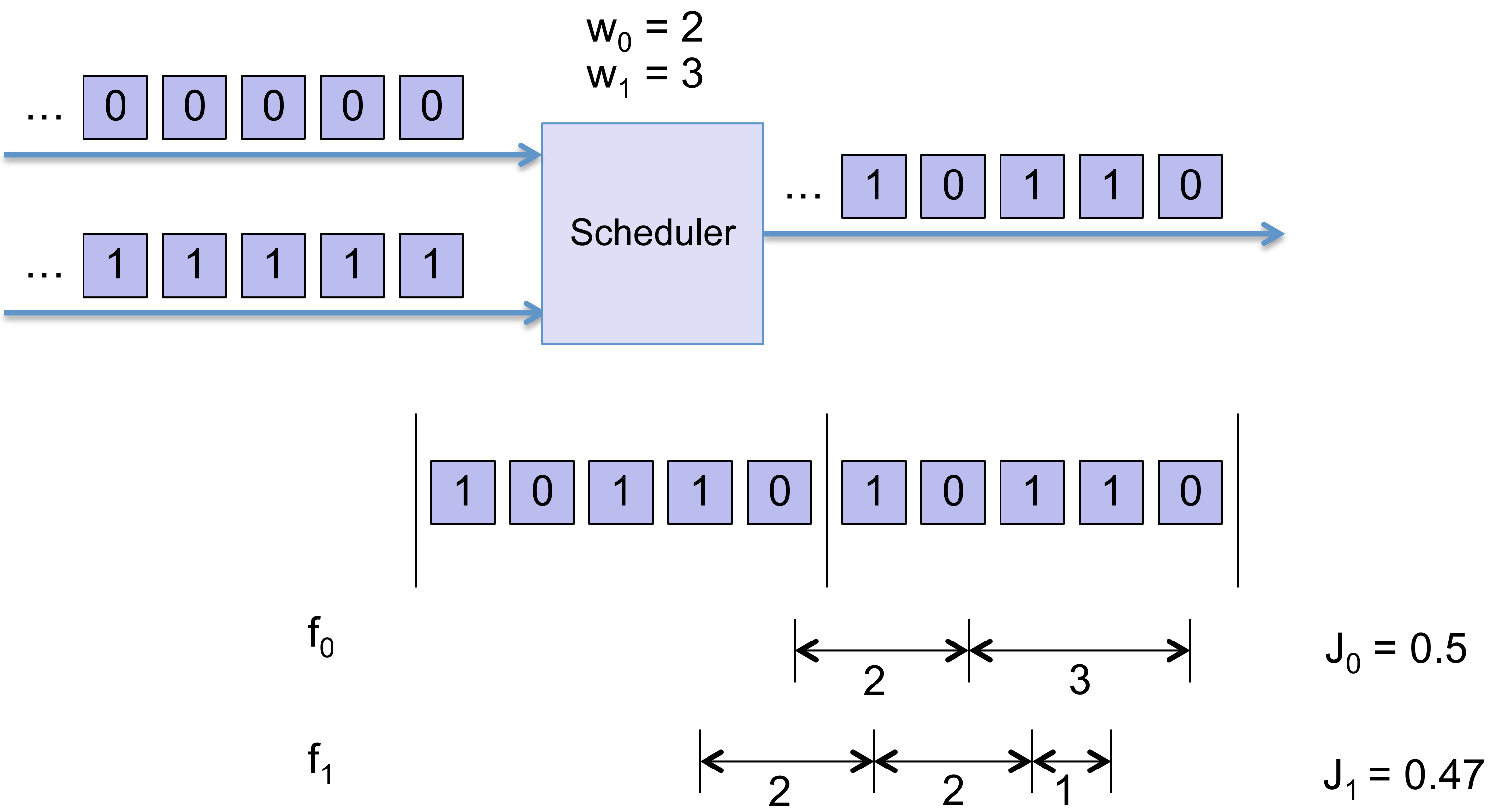


Scheduling > 2 Flows – Non-Iterative Method

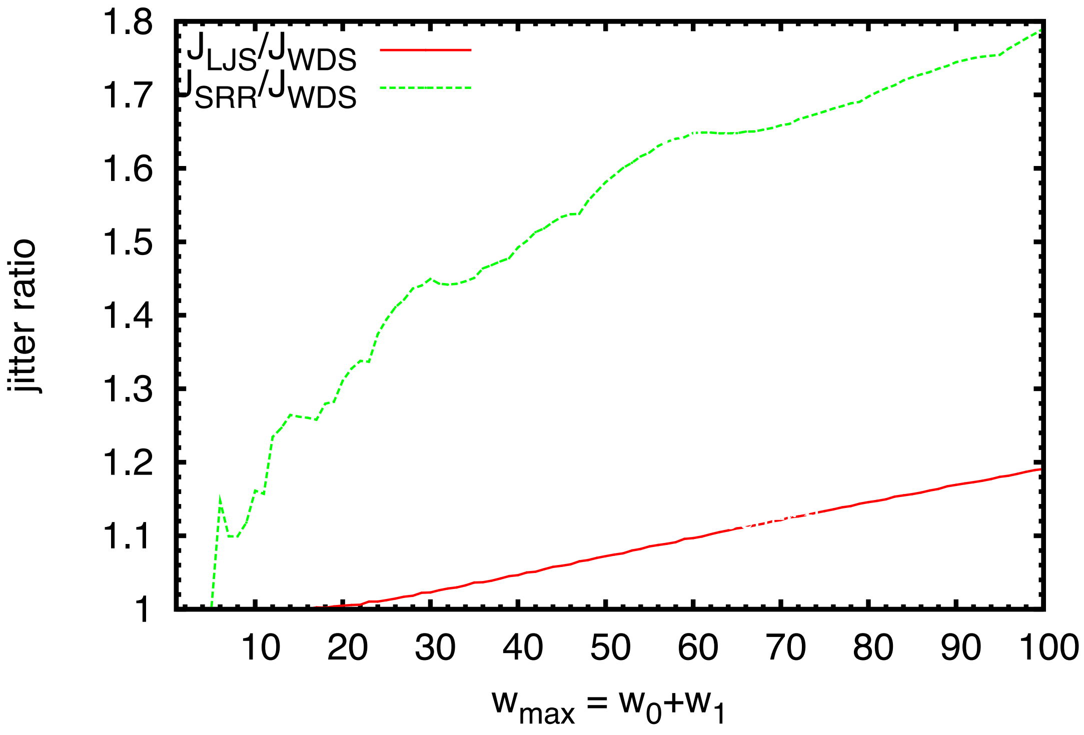
```
err1 $\Leftarrow$ 0; err2 $\Leftarrow$ 0
deltaerr1 $\Leftarrow$ w1/w0; deltaerr2 $\Leftarrow$ w2/w0
**loop**
    serve f0
    err1 $\Leftarrow$ err1 + deltaerr1; err2 $\Leftarrow$ err2 + deltaerr2
    **while** err1 $\geq$ 0.5 or err2 $\geq$ 0.5 **do**
        **if** err1 $\geq$ 0.5 **then**
            serve f1; err1 $\Leftarrow$ err1 – 1
        **end if**
        **if** err2 $\geq$ 0.5 **then**
            serve f2; err2 $\Leftarrow$ err2 – 1
        **end if**
    **end while**
  **end loop**
```

Jitter Calculation

Jitter = stdev(Interdeparture Delays)



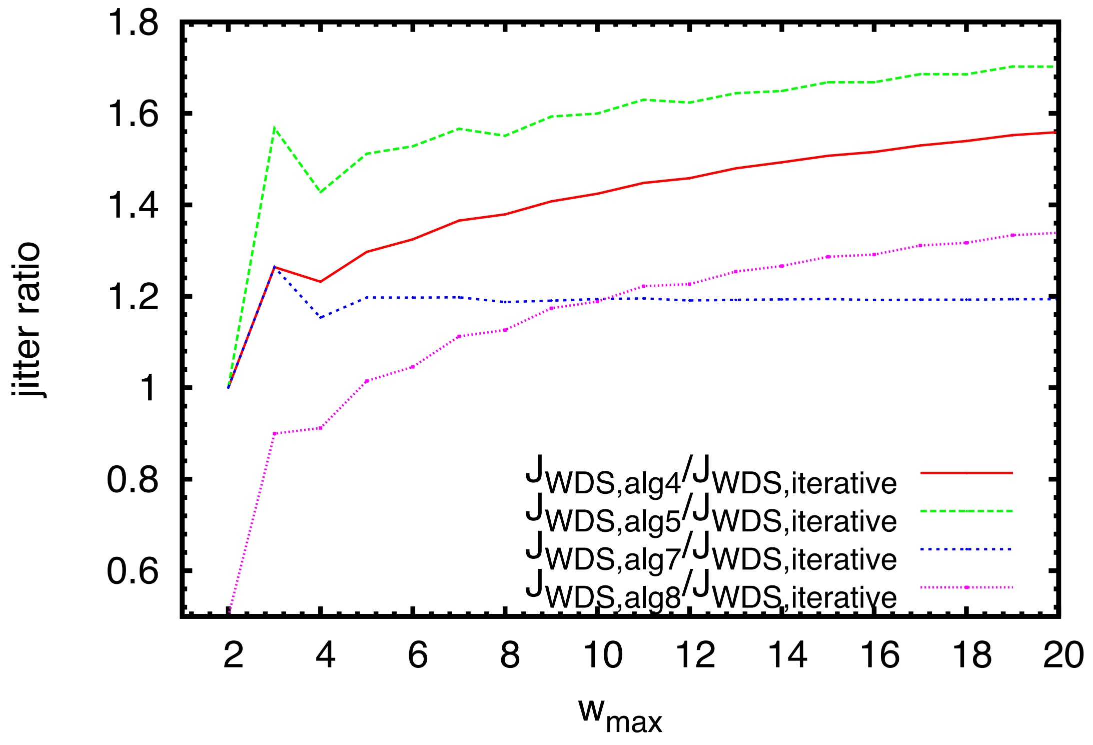
Results: Jitter for 2 Flows



Algorithms:

- WDS = Weighted Differential Scheduler
- LJS = Low Jitter Scheduler
- SRR = Smooth Round Robin Scheduler

Results: Jitter for 3 Flows



- Alg. 4: Increment error only when packet from f_0 is forwarded
- Alg. 5: Increment error when packet from f_0 or f_1 is forwarded
- Alg. 7: Treat flows symmetrically
- Alg. 8: Serve flow with biggest error first

Summary

- Packet scheduling can be described as a Digital Differential Analyzer
- WDS has lower jitter than any other known weighted scheduler
- WDS requires integer additions/subtractions only
- Future work: better understand and improve non-iterative WDS versions

ORACLE®

hans.eberle@oracle.com

wladek.olesinski@oracle.com