Parallelization in Action with SAS Analytic Procedures

Robert Cohen
Senior Research Statistician
Linear Models R&D
Your Rise and Shine Menu

- Parallelization adds value to the IVC

**Marketing:** I should have slept in
- Multithreading to provide parallel execution

**Boring:** I should have left when I had the chance
- How do you measure scalability

**Insulting:** This guy thinks I’m a 10 year old
- Selected demonstrations

**Deceiving:** The truth, but not the whole truth
IVC: Parallelization Adds Value

- Complete today’s analyses faster
- Analyze tomorrow’s problems within today’s time constraints
The IVC in Action
Changes You Have to Make in Your Legacy Code

TINSTAAFL
Unthreaded GLM: 2 CPU Box

- GLM runs in a single thread
- GLM never blocks this thread
- GLM work is NOT done in parallel
Unthreaded GLM: 2 CPU Box

Thread View:  Running | Waiting I/O | Blocked | Exited

CPU Utilization:  --- CPU 1 --- CPU 2 ---
Unthreaded GLM: 2 CPU Box

Thread View:  
- Running
- Waiting I/O
- Blocked
- Exited

Combined CPU Utilization

CPU Usage History

0 50 100
Multithreaded GLM: 1 Active Thread
2 CPU Box

- Worker threads used for specific tasks
- GLM thread blocks while a worker thread is active
- GLM does not execute in parallel
Multithreaded GLM: 1 Active Thread
2 CPU Box

Thread View:
- Running
- Waiting I/O
- Blocked
- Exited

CPU Utilization:
- CPU 1
- CPU 2
Multithreaded GLM: 1 Active Thread
2 CPU Box

Thread View: Running, Waiting I/O, Blocked, Exited

Combined CPU Utilization
Multithreaded GLM: 2 Active Threads
2 CPU Box

- GLM thread spawns off worker threads
- Two independent worker threads per task
- Work is done in parallel
Multithreaded GLM: 2 Active Threads
2 CPU Box

Thread View: Running, Waiting I/O, Blocked, Exited

CPU Utilization: CPU 1, CPU 2

Copyright © 2003, SAS Institute Inc. All rights reserved.
Multithreaded GLM: 2 Active Threads
2 CPU Box

Combined CPU Utilization
Multithreaded GLM: 4 Active Threads
2 CPU Box

<table>
<thead>
<tr>
<th>Thread View:</th>
<th>Running</th>
<th>Waiting I/O</th>
<th>Blocked</th>
<th>Exited</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Threading Comparison
Multithreaded GLM: 2 CPU Box

Thread View:  Running  Waiting I/O  Blocked  Exited

[Thread View Diagram]
## Amdahl's Law

### PF = 80%

<table>
<thead>
<tr>
<th>CPUs</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1.67</td>
</tr>
<tr>
<td>4</td>
<td>2.50</td>
</tr>
<tr>
<td>8</td>
<td>3.33</td>
</tr>
<tr>
<td>16</td>
<td>4.00</td>
</tr>
<tr>
<td>32</td>
<td>4.44</td>
</tr>
</tbody>
</table>

**Scalability Analysis**

- **Not Scalable**
- **Scalable**

**Legend:**
- Blue: Not Scalable
- Yellow: Scalable
Amdahl’s Law

Parallelizable Fraction

- 100%
- 99%
- 95%
- 90%
- 80%
- 60%

CPU Usage vs. Speedup
Scalability in PROC REG: Wide Data and Scalar I/O

Test Details

50,000 observations
500 predictors
Stepwise Selection
Scalar I/O

Speedup vs. CPUs Used

- Linear
- Amdahl, PF=93%
Scalability in PROC REG: Wide Data and Scalar I/O

Test Details
50,000 observations
500 predictors
Stepwise Selection
Scalar I/O

Speedups
- Linear
- Amdahl, PF=93%
- Achieved
Scalability in PROC REG: Narrow Data, Parallel I/O

Test Details
4 million observations
20 predictors
Parallel I/O

Diagram showing speedup vs. CPUs used with linear and Amdahl's law speedup lines, indicating linear scalability and an efficiency of 99.9%.

Speedups
- Linear
- Amdahl, PF=99.9%
Scalability in PROC REG: Narrow Data, Parallel I/O

Test Details
- 4 million observations
- 20 predictors
- Parallel I/O
Scalability in PROC DMREG

Test Details

500,000 observations

Predictors:
- 50 continuous
- 15 classification

Logistic model

Parallel I/O

Graph showing speedups with increasing CPUs used. The graph compares linear speedup to Amdahl's Law with a performance factor of 93%.
Scalability in PROC DMREG

Test Details

- 500,000 observations
- Predictors:
  - 50 continuous
  - 15 classification
- Logistic model
- Parallel I/O

![Graph showing speedup with CPUs used, indicating linear and Amdahl speedup with a parallel I/O evaluation.]

- Speedups:
  - Linear
  - Amdahl, PF=93%
  - Achieved
Baseline Speedup and Scalability in PROC DMREG

Test Details
500,000 observations
Predictors:
  50 continuous
  15 classification
Logistic model
Parallel I/O

Speedups
- Linear
- Amdahl, PF = 93%
- Achieved
- V9/V8

* * *
Scalability in PROC GLM

Test Details

6000 observations
4 classification variables
2000 parameters

Speedups

- Linear
- Amdahl, PF = 98%
Scalability in PROC GLM

Test Details

6000 observations
4 classification variables
2000 parameters

Superlinear Scalability!
Scalability in PROC LOESS

Test Details

4000 observations
18 models evaluated
Confidence limits for selected model

![Graph showing speedups with CPUs Used]
Scalability in PROC LOESS

**Test Details**

- 4000 observations
- 18 models evaluated
- Confidence limits for selected model

**Graph Details**

- **Speedups**
  - Linear
  - Amdahl, PF=95%
  - Achieved

- **Axes**
  - X-axis: CPUs Used
  - Y-axis: Speedup

- The graph shows a linear relationship between the number of CPUs used and the speedup achieved, with data points closely following a linear trend for 1 to 8 CPUs.
Scalability in PROC LOESS

Test Details

4000 observations
1 model specified
Confidence limits for specified model
Scalability in PROC LOESS

Test Details

4000 observations

1 model specified

Confidence limits for specified model

Speedups

- Linear
- Amdahl, PF=99%
- Achieved
Partially Multithreaded Procedures

- **Base SAS**
  - PROC SORT
  - PROC SUMMARY
  - SQL (Group by, Order by)

- **SAS/STAT**
  - PROC GLM
  - PROC LOESS
  - PROC REG
  - PROC ROBUSTREG

- **Enterprise Miner**
  - PROC DMDB
  - PROC DMREG
  - PROC DMINE

**NOTE:** Not all usages of these procedures are scalable.

Your mileage may vary!
Reading Between the Lines

- Parallelization adds value to the IVC

Analyze bigger volumes of data
- Multithreading to provide parallel execution

Not as boring as I feared
- How do you measure scalability

Predicting scalability is a subtle task
- Selected demonstrations

Some of my jobs will run faster in SAS 9
Questions

and hopefully answers