Analysis and Tuning of IMS Scheduling Using PWFI

Rich Lewis
IBM Dallas Systems Center

Abstract and Trademarks

Abstract

Scheduling in IMS Transaction Manager systems depends on many IMS parameters. These parameters may have significant effects on the resources used to schedule transactions. This presentation provides an overview of scheduling and a review of scheduling parameters such as PARLIM, PROCLIM, WFI, and PWFI. It also presents a scheme for minimizing scheduling overhead by using Pseudo Wait-for-Input (PWFI). This includes advice on calculating PARLIM and PROCLIM values, assigning classes, and determining the number of regions to use for each class. The presentation is based on IMS/ESA 5.1 functions.

Agenda

- Overview of Scheduling
- Parameters that Affect Scheduling
  - PARLIM, MAXRGN, PROCLIM, WFI, PWFI
- A Scheme for Minimizing Scheduling Overhead
- Steps to Implement the Scheme

Scheduling Overview

- **IMS actions associated with scheduling**
  - **Scheduling pathlength**
  - **Scheduling serialization (latching)**
  - **Program loading (LLA/VLF, Preload, ...)**
  - **Program housekeeping**
  - **Subroutine loading**
  - **DB2 thread creation/termination !!!**

Scheduling Overhead

- **"Measuring" overhead with the IMS Monitor**
  - **DB2 Thread Creation/Termination**
    - CREATE THRD and TERM THRD in Region IWAIT, Program I/O, and Call Summary reports
  - **Scheduling pathlength and serialization**
    - Mean times in Scheduling + Termination sections of Region Summary reports
  - **Program load and program housekeeping**
    - Sched. to 1st call/Sched. in Program Summary reports

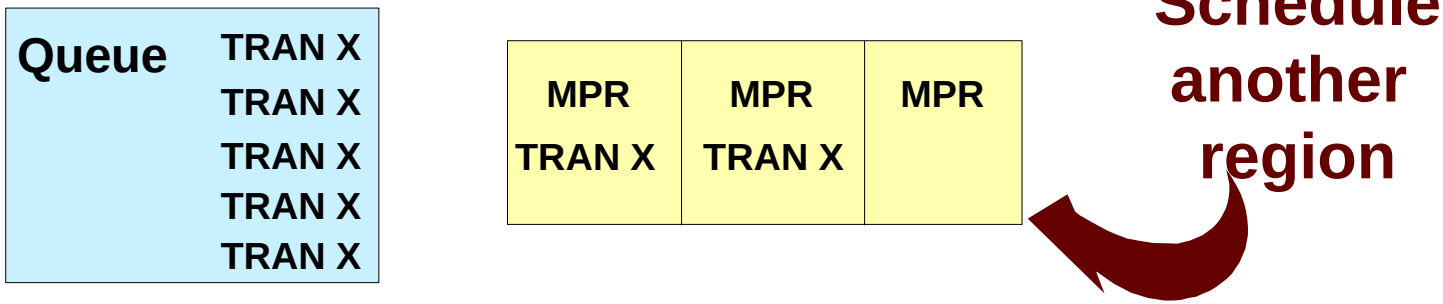
Scheduling Parameters

- PARLIM
- MAXRGN
- PROCLIM
- WFI
- PWFI

PARLIM Parameter

- **PARLIM Parameter on TRANSACT macro**
  - **Requires SCHDTYP=PARALLEL on APPLCTN macro**
  - **Determines if transaction will be scheduled in *another* region**
    - If number on queue is greater than PARLIM times number of regions already scheduled, schedule another region

```
TRANSACT
 CODE=X,
 PARLIM=2
```



Choosing a value for PARLIM

- **Set PARLIM to minimize scheduling overhead while providing necessary internal response time**

- **Response time =**
  - **Time waiting on queue**
    - <= PARLIM times avg. execution time

    or

    Scheduling time

  plus
  - **Execution time**
    - Avg. execution time

Choosing a value for PARLIM

- If PARLIM is 'too low' we unnecessarily schedule programs which could have waited for an already scheduled region
  - We waste our resources for no benefit!

Choosing a value for PARLIM ...

- Resp. Time = (PARLIM x Avg. Exec. Time)
  + Avg. Exec. Time
- Solving for PARLIM:

  PARLIM = (Resp. Time / Avg. Exec. Time) - 1

Acceptable response time = 1 sec.
Avg. execution time = 0.2 sec.
PARLIM = (1 / 0.2) - 1 = 4

Acceptable response time = 2 sec.
Avg. execution time = 0.3 sec.
PARLIM = (2 / 0.3) - 1 = 5

Acceptable response time = 2 sec.
Avg. execution time = 0.1 sec.
PARLIM = (2 / 0.1) - 1 = 19

MAXRGN Parameter

- **MAXRGN parameter on TRANSACT macro**

```
TRANSACT ...MAXRGN=N,...
```

  - **Limits the number of MPP regions that may have the transaction concurrently scheduled**
  - **Of course, the number is also limited by the number of MPP regions which are eligible to schedule the transaction's class**
  - **Often used to keep a "dog" transaction from occupying too many regions**

PROCLIM Parameter

- **PROCLIM parameter on TRANSACT macro**

  `TRANSACT ...PROCLIM=(count,time),...`

  - **Count determines if IMS will attempt to schedule another transaction in region when there are more of this transaction on the queue**
    - 65535 eliminates limit on # trans. processed
  - **If number of transactions processed by the region equals PROCLIM count, IMS may attempt to schedule another transaction in region**
  - **Used to get a 'low priority' transaction out of region so a 'high priority' transaction may be processed**
    - High priority transactions should use 65535

PROCLIM Parameter (cont.)

- **Quick Reschedule**
  - **Avoids unnecessary terminations and schedulings**
  - **Invoked when PROCLIM count > 0**
  - **When PROCLIM count reached**
    - If same transaction would be rescheduled
      - Does not return 'QC' status code
      - Returns next message
    - If same transaction would not be rescheduled
      - Returns 'QC' status code
      - Program terminates
  - **If PROCLIM count = 0**
    - Quick Reschedule is disabled
    - Only one transaction processed per schedule

Wait for Input (WFI)

- **What is WFI?**
  - **Specified on the system definition TRANSACT macro**

```
TRANSACT CODE=X,...,WFI,...
```

  - **Lack of message does not cause IMS to terminate program in region**
    - GU to IO-PCB will not return 'QC' status code
    - Instead, waits for next message to arrive for the transaction
  - **Used to avoid scheduling overhead**

Wait for Input (WFI) (cont.)

- **Operations with WFI**
  - **WFI transactions do not give up their regions**
  - **Program terminated only by**
    - Reaching PROCLIM
    - `/STOP TRANSACTION` for MPP
    - `/PSTOP REGION reg# TRANSACTION tran` for BMP
    - Shutdown checkpoint of IMS system
  - **There must be enough regions for peak demand**

    or
  - **Operations must start and stop regions as required**

Pseudo Wait for Input (PWFI)

- **What is PWFI?**
  - **Specified for the MPP region on JCL**

```
// EXEC PGM=DFSRRC00,PARM='MSG,...,PWFI=Y,...'
```

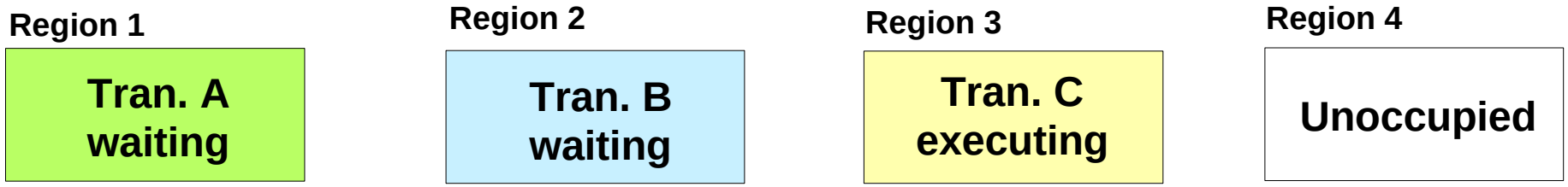
  - **Determines what IMS will do when there is nothing for a region to do:**
    - GU to IO-PCB will not return 'QC' status code
    - Instead, waits for next message to arrive for the region
    - If next message is the same transaction code, avoids termination and scheduling of transaction
    - If next message is different transaction code, returns 'QC', terminates program, and schedules new one

PWFI is Self Tuning

- Transaction tends to be scheduled in region which already has the tran. code scheduled
  - In PWFI wait
- If not already scheduled, tran. tends to be scheduled in unscheduled region
  - Region not waiting and eligible for this class
- If tran. not already scheduled and all eligible regions are scheduled, region which has been idle longest is used
  - All regions for this class are scheduled

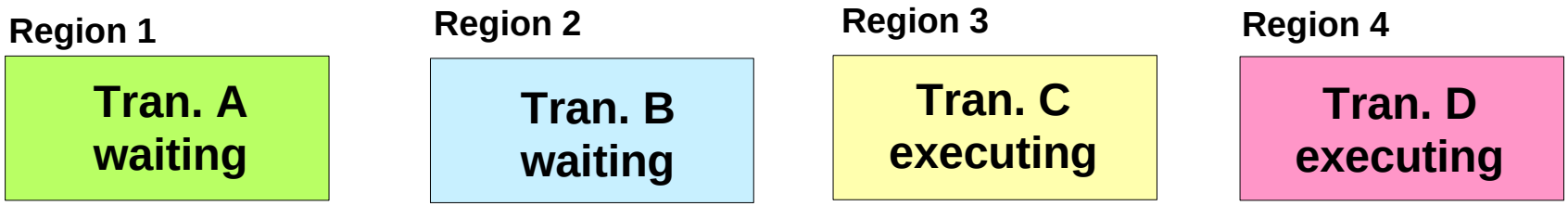
Self Tuning Example

- **If next tran is:**
  - **A: given to region 1**
  - **B: given to region 2**
  - **C: waits for region 3 or given to region 4**
    - Depends on PARLIM, number on queue, etc.
  - **D: schedules in region 4**



Self Tuning Example

- **If next tran. is:**
  - **A: given to region 1**
  - **B: given to region 2**
  - **C: waits for region 3 or given to region 1 or 2**
    - Depends on PARLIM, number on queue, etc.
  - **D: waits for region 4 or given to region 1 or 2**
    - Depends on PARLIM, number on queue, etc.
  - **E: schedules in region 1 or 2**
    - Schedules in region that has been waiting the longest



Self Tuning Rule

A simple rule:

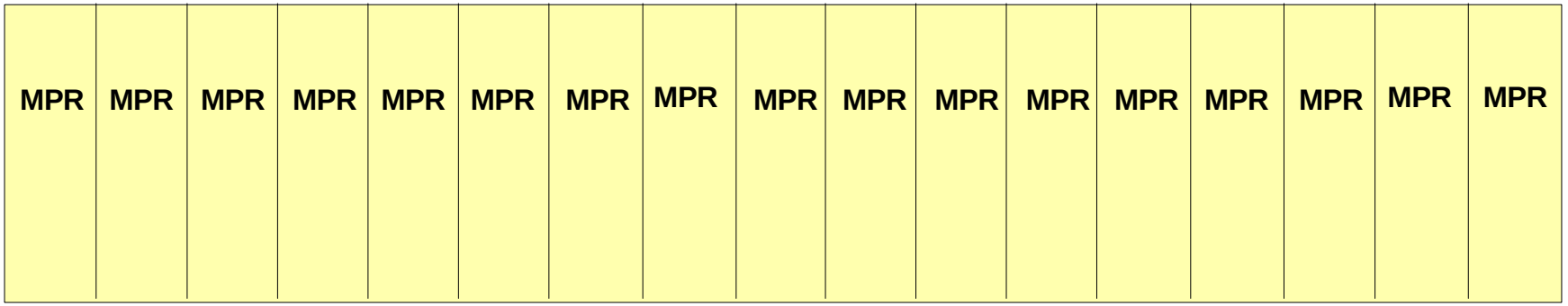
When all regions are PWFI and a transaction is x% of the volume, it will tend to occupy x% of the message regions for its class.

Implication:

The more PWFI regions we have, the more scheduling overhead we avoid.

A Question

Q: Can we do better than just having a lot of regions with one class?



A: It depends on our volumes by transaction.

A Better Scheme

- Give each high-volume transaction its own class and its own dedicated region(s)
- Put low-volume transactions in 'shared' regions
- Use 'shared' regions for 'overflow' of high-volume transactions

Tran X Class 1	Tran X Class 1	Tran Y Class 2	Tran Z Class 3	Classes 4 and 1	Classes 4 and 2	Classes 4 and 3	Class 4	Class 4

A Better Scheme (cont.)

- **Give each high-volume transaction its own class and its own dedicated region(s)**
  - **Use PWFI for the regions**
  - **Regions have only one class assigned to them**
  - **Eliminates scheduling for dedicated regions**
  - **Avoids having other transactions 'steal' the region(s)**

Tran X Class 1	Tran X Class 1	Tran Y Class 2	Tran Z Class 3	Classes 4 and 1	Classes 4 and 2	Classes 4 and 3	Class 4	Class 4

|---------- Dedicated Regions ----------|

A Better Scheme (cont.)

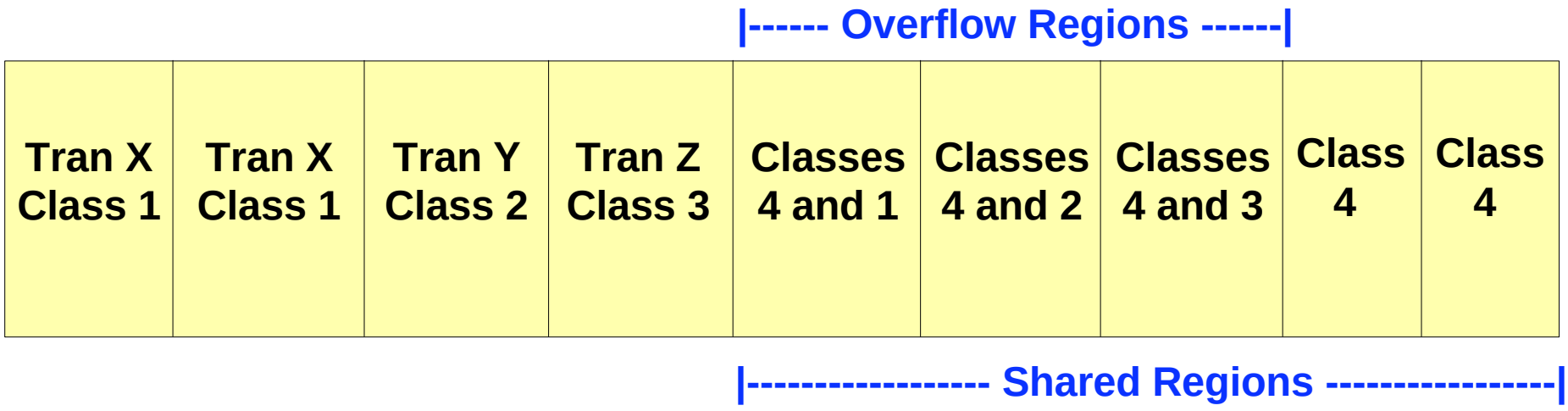
- **Put low-volume transactions in 'shared' regions**
  - **Use one (or few) classes for them**
    - This example uses class 4 for low-volume transactions

Tran X Class 1	Tran X Class 1	Tran Y Class 2	Tran Z Class 3	Classes 4 and 1	Classes 4 and 2	Classes 4 and 3	Class 4	Class 4

|---------- Shared Regions ----------|

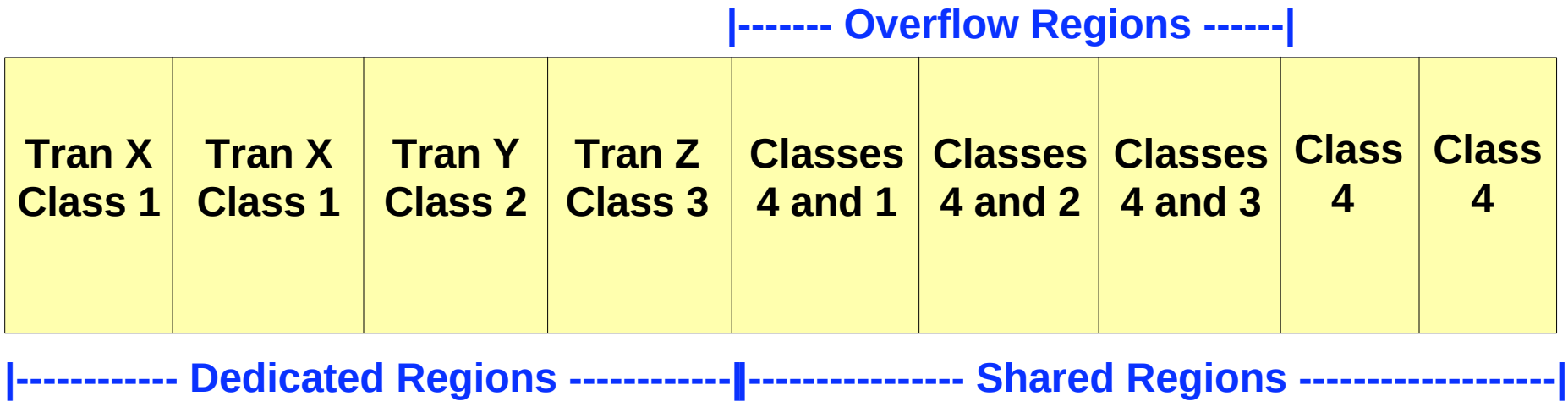
A Better Scheme (cont.)

- **Use 'shared' regions for 'overflow' of high-volume transactions**
  - **Overflow occurs when volumes are so high that the dedicated region(s) are not enough**
  - **This is not practical with WFI**
    - WFI would never release the regions



A Better Scheme (cont.)

- For each high volume transaction
  - Must know how many regions are needed
    - Dedicated
    - Overflow



The Number of Regions for a Tran.

- **How many dedicated regions are needed for a tran.?**
  - **Enough to meet typical needs**
  - **# Regions = (Typical tran. rate x Avg. exec. time) / Region occupancy**
    - Example:

      Typical tran. rate = 2 per second

      Avg. exec time = 0.4 seconds

      Region occupancy = 0.65

      # regions = (2 x 0.4) / 0.65 = 1.23 = 2 (rounding up)

The Number of Regions for a Tran.

- **How many overflow regions are needed for a transaction?**
  - **Total regions minus dedicated regions**
- **How many total regions are need for a transaction?**
  - **Enough to meet peak needs**
  - **# Regions = (Peak tran rate x Avg. exec. time) / Region occupancy**
    - Example:

      Peak tran. rate = 5 per second

      Avg. exec time = 0.4 seconds

      Region occupancy = 0.65

      # regions = (5 x 0.4) / 0.65 = 3

Steps to Implement the Scheme

1. Order transactions by volume
2. Adjust transactions for cost of scheduling
3. Select transactions for special classes
4. Calculate number of dedicated regions
5. Calculate number of overflow regions
6. Assign classes, PARLIM, and PROCLIM values
7. Create dependent regions with new classes

Steps to Implement the Scheme

Step 1. Order transactions by volume

Source of information:

- IMS Monitor Transaction Queuing report
- IMSPARS
- IMS Log records
- Installation reports

Steps to Implement the Scheme

Step 2. Adjust transactions for cost of scheduling

Sources of information:

- IMS Monitor
  - Call Summary report or
  - Program I/O report
    Create Thrd and Term. Thrd
  - Program Summary report
    Sched. to First Call

Adjust transaction volumes by giving extra weight to transactions using DB2 and/or having high 'Sched. to First Call' times

Steps to Implement the Scheme

Step 3. Select transactions for special classes

a. Reorder transactions after weighting for DB2 and 'Sched. to First Call'

b. Eliminate transactions whose volumes are less than 1% of all transaction volumes

c. Select up to 20 transactions

Steps to Implement the Scheme

Step 4. Calculate number of dedicated regions

Regions = (Typical tran. rate x Avg. exec. time)/
Region occupancy

Typical tran. rate - IMS Monitor Transaction Queuing report

Avg. exec. time - IMS Monitor Program Summary report

Region occupancy - 65% (rule of thumb)

Steps to Implement the Scheme

Step 5. Calculate number of overflow regions

Regions = (Peak tran. rate x Avg. exec. time)/
Region occupancy

Typical tran. rate - IMS Monitor Transaction Queuing report from peak time
or
2 times average rate (rule of thumb)

Avg. exec. time - IMS Monitor Program Summary report

Region occupancy - 65% (rule of thumb)

Steps to Implement the Scheme

Step 6. Assign classes, PARLIM, and PROCLIM values

For selected transactions,

a. Assign unique classes on their TRANSACT macros

b. Calculate PARLIM values

PARLIM = (Response time/Avg. Exec. Time) - 1

Response time = 1 to 2 seconds (installation dependent)

Avg. Exec. Time - IMS Monitor Program Summary report

c. Set PROCLIM = 65535

Steps to Implement the Scheme

Step 7. Create dependent regions with new classes

For selected transactions,

a. Create dedicated regions
   - Each will have one new class

b. Create overflow regions
   - Each will have a shared class and one new class

For other transactions,

a. Create or maintain shared regions
   - Each will have one or more shared classes

Benefits of the Scheme

- Avoids unnecessary overhead
- Dynamically adjusts to changing volumes
- Provides the required response times

Summary

- **Scheduling may produce unnecessary overhead**
  - **Especially with DB2 thread creation/termination**
- **Scheduling overhead may be controlled**
  - **PWFI may be used to minimize overhead**
  - **PARLIM may be calculated**
  - **Numbers of regions may be calculated**
- **Data is available for the calculations**