# Turbo Dispatcher for the Real World

## Session E77

Dan Janda
VSE/ESA System Center
Endicott, New York

May, 2000



RETURN TO INDEX

# Contents

- Turbo Dispatcher Background
  - Why have a Turbo Dispatcher
  - Multiprocessor Considerations
- Partition Considerations
- Technical Details
- Exploiting Turbo Benefits
- Summary
  - Are You Turbo Ready?

# Abstract and Copyright Notice


- Abstract: The VSE/ESA Turbo Dispatcher brings the power of multi-engine processors to the native VSE user, and brings the ability to give a VSE/ESA guest of VM/ESA the ability to use more than the power of a single engine to process its workload. As new processors are developed, the multi-processor technology permits IBM to deliver more throughput at a lower cost than would be available if a single processor had to deliver all the power needed for that workload. In addition, the Turbo Dispatcher gives new priority scheduling capabilities to the VSE user in terms of balancing the shares of CPU power.

  This presentation will present the Turbo Paradigm, including an overview of Single- and Mulit-Processor environments and their workloads. The environmental factors where multi-processors shine, and those which make single processors shine even brighter will be discussed. Finally, what you should look for in your environment, and what you can change in your environment to exploit the Turbo Dispatcher will be described.




# Bibliography, Trademarks and Disclaimer

- For more information about the subject of this presentation, refer to:

  | | |
  |---|---|
  | "VSE/ESA System Control Statements" | SC33-6613 |
  | "VSE/ESA Turbo Dispatcher Guide" | SC33-6599 |
  | "VSE/ESA Turbo Dispatcher Guide and Ref." | SC33-6797 (2.4) |
  | "VSE/ESA Version 2 Release 1 Turbo Dispatcher" | SG24-4674 |
  | "VSE/ESA 2.1 Performance Documents" | VE21PERF PACKAGE |

  (available from the IBM VSE/ESA Home Page on the Internet --
  http://www.s390.ibm.com/vse/
- The following words used in this presentation are trademarks or registered trademarks of the IBM Corporation:

  | | | | | | |
  |---|---|---|---|---|---|
  | VSE | VSE/ESA | ESA | VSE/VSAM | CICS | CICS/VSE |
  | ECKD | ESA/370 | ESA/390 | PR/SM | VM/ESA | VTAM |

- This presentation contains information gathered from laboratory tests and from specific customer experiences. Applicability of these results in any other environment is the responsibility of the user, and cannot be guaranteed.

# Turbo Dispatcher Background

- Why have a Turbo Dispatcher?
  - **To get more throughput for VSE users**
  - **To allow VSE users to exploit current processor technology**

# More Throughput?

- Workloads are growing
- More and New Applications are coming
- New technologies are adding work to our mainframes:
  - e-Business and the World-Wide Web
  - TCP/IP
  - ...

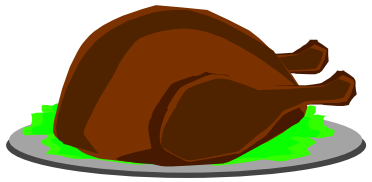
# Exploit Current Processor Technology?

- Fast uni-processors cost more than equivalent throughput multi-processors
- For growth, it may be less expensive to use multi-processors
  - IBM 9672 Enterprise Servers
    - 1 to 12 engines per processor
  - IBM 2003 Multiprise Servers
    - 1 to 6 engines per processor

# Multi-Processor Factors

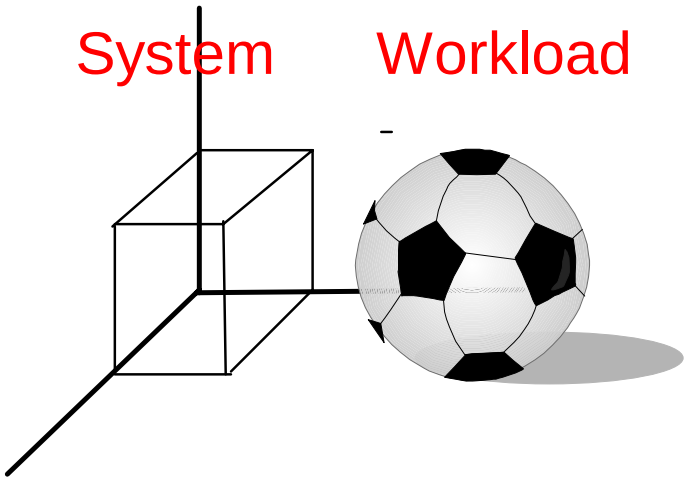
- Traditional Constraints
  - CPU Cycles
  - I/O Bandwidth
  - Storage
- Additional Constraints experienced migrating to n-ways

(What's a heN weigh, anyway?)



# Traditional Constraints

- CPU Cycles
- I/O Bandwidth
- Storage




- Each can be traded off against the others
- They need to be increased in balance for optimum economic throughput
- Optimum balance points change with time

# Turbo CPU Constraints

- Total processor power for total work
- Single engine power for single partition
- Single engine power for non-parallel part of workload
- Sufficient workload to occupy all engines

# Turbo CPU Constraints - Total Processor Power

- Total power of all engines must exceed the needs of entire workload at its peak times
  - This is not a new constraint
  - No different than with uni-processors
  - **Do Not Forget:**
    - *Workload Growth!*
    - *Additional CPU Load from new system software upgrades!*

# Turbo CPU Constraints - Single Engine - Single Job

- A single VSE job or partition can only use the power of one engine
  - Example - 9121-511 to 2003-224 (Total ratio 35::43, Engine ratio 35::21)
    - High Priority jobs are limited to power of one engine - less power available to them
    - Low Priority jobs get more power
    - *What's important? What's high priority?*
    - ***Any single job will run slower!***

# Turbo CPU Constraints - Non-parallel components

- The non-parallel part of the entire workload must be able to be handled by a single engine
  - QUERY TD command shows NP Share
  - Dispatcher will search for parallel work when parallel resource limits throughput
  - CPU Utilization (overhead) increases with no increase in throughput
  - Similar to MVS' "low utilization effect" -- does not reduce total available capacity

# Turbo CPU Constraints - Sufficient Workload

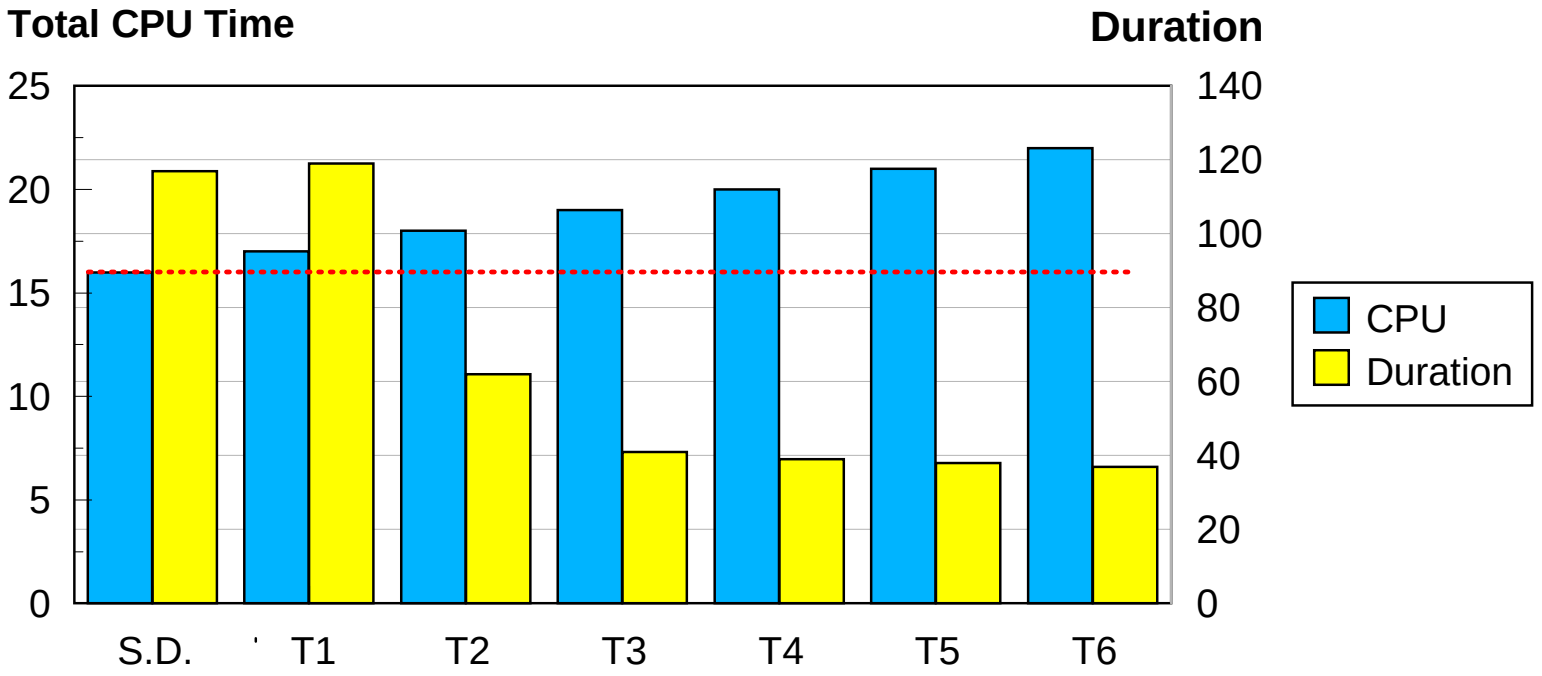
- Even with totally CPU workloads
  - There must be as many partitions ready to run as there are processors
  - Lost cycles are LOST -- they cannot help a job running on another processor
- An 3-way can run 3 CPU jobs in about the same time as 1 CPU job. A 4-way will not speed up this 3 job workload.

# Turbo CPU Constraints- Speed vs. Throughput

- A highway represents a good analogy:
  - With little traffic (workload) and the same speed limit, a single lane road will get you to your destination in the same time as a multi-lane expressway, but as traffic increases the expressway handles it better. A single lane road with a higher speed limit may get you there quicker, but the expressway can move more traffic less expensively.

# Turbo CPU Constraints- Speed vs. Throughput...

- Increased throughput, reduced elapsed time




NPS ~ 0.10 -- 9 engines
(not characteristic of most real workloads)
High CPU intensity, very low I/O content

# Turbo Dispatcher - Partition Considerations

- I should not expect my speed-limited job to run faster -- but I'll get more throughput if I have the workload
- VSE Scheduling considerations
  - The Standard (former) Dispatcher
  - The Turbo Dispatcher

# T D - Partition Scheduling

- The VSE Standard Dispatcher was (is) a "Preempt-Resume" Priority Scheduler
  - Strict sequence of priorities
  - Highest priority ready task runs until:
    - it is preempted (via an interrupt) for a higher priority task that has become ready
    - it voluntarily waits
  - After preemption, it will run again when it is the then highest priority ready task

# T D - Partition Scheduling...

- The VSE Standard Dispatcher was modified to support time-sliced balancing
  - A single group of balanced partitions
  - At the end of a balancing interval (MSECS), the CPU used by each balanced partition during the interval is computed and used to adjust the relative priorities of the partitions for the next interval.
  - Dynamic Partition Classes were weighted the same as each individual static partitions

# T D - Partition Scheduling

- The VSE Turbo Dispatcher is also a "Preempt-Resume" Priority Scheduler
  - It can handle scheduling of multiple CPU "engines"
  - It still uses the same priority structure
  - It adds significant power and control for "balancing" or "time slicing"

# T D - Partition Scheduling...

- The VSE Turbo Dispatcher was built to support time-sliced balancing among both dynamic and static partitions
  - At the end of a balancing interval (MSECS), the CPU used by each balanced partition during the interval is computed and used to adjust the relative priorities of the partitions for the next interval.
  - Dynamic Partitions (not Classes) are weighted the same as each individual static partitions

# T D - Partition Scheduling...

- The VSE Turbo Dispatcher also provides Relative Share balancing
  - When a partition uses its allocated share of CPU time, its priority (within the balanced group) is lowered relative to the other group partitions
  - Dynamic Partitions (not Classes) are weighted the same as each individual static partitions
  - This allows you to prevent a high priority partition from monopolizing resources within the balanced group.

# T D - Commands - control:

**PRTY**

```
 PRTY BG,C,FB,FA,F5,F9,F8,F7,F6,F4,F2,F3,F1
```

(Assume POWER in F1, CICS in F2, VTAM in F3)

**PRTY F4=F8=C=F2,BELOW,F3**

```
 PRTY BG,FB,FA,F5,F9,F7,F6,F4=F8=C=F2,F3,F1
 SHARE F4= 100, F8= 100,  C= 100, F2= 100
```

(Define a balance group and set it in priority below F3.
Assume 2 active class C partitions, total share=500)

**PRTY SHARE,F2=600**

```
 PRTY BG,FB,FA,F5,F9,F7,F6,F4=F8=C=F2,F3,F1
 SHARE F4= 100, F8= 100,  C= 100, F2= 600
```

(Increase the share value for F2 to 60%)

# T D - Commands - control...

**`PRTY SHARE,F4=10,F8=10,C=10,F2=60`**

```
 PRTY BG,FB,FA,F5,F9,F7,F6,F4=F8=C=F2,F3,F1
 SHARE F4=  10, F8=  10,  C=  10, F2=  60
```

(same effect as previous command, as settings are relative)

**`PRTY FB,EQUAL,F2`**

```
 PRTY BG,FA,F5,F9,F7,F6,FB=F4=F8=C=F2,F3,F1
 SHARE FB= 100, F4= 10, F8= 10, C= 10, F2= 60
```

(FB has default value of 100, total is 200,
so FB gets 50%, F2 gets 30%)

# T D - Commands - control...

**In startup procedure $0JCL you can include:**

```
SYSDEF TD,START=[ALL|cpuaddr]
```

to start additional CPUs

**At the VSE/ESA console, you can enter AR commands:**

```
SYSDEF TD,START=[ALL|cpuaddr]
```

or

```
SYSDEF TD,STOP=[ALL|cpuaddr]
```

or

```
SYSDEF TD,STOPQ=[ALL|cpuaddr]
```

or

```
SYSDEF TD,RESETCNT
```

# T D - Commands - monitoring

```
QUERY TD[,INTERNAL]
 CPU    STATUS   SPIN_TIME   NP_TIME   TOTAL_TIME NP/TOT
  00    ACTIVE        5656     40856       147691  0.276
  01    ACTIVE        4675     39930       148688  0.268
  02  INACTIVE
  03    ACTIVE        4619     39940       148734  0.268
      ----------------------------------------------------
TOTAL                14950    120726       445113  0.271

                NP/TOT: 0.271  SPIN/(SPIN+TOT): 0.032
 OVERALL UTILIZATION: 294%     NP UTILIZATION:  77%

 ELAPSED TIME SINCE LAST RESET:   156094
```

(All time values are in milliseconds - 3600000 = 1 hour

**SYSDEF TD,RESETCNT** can be used to define interval beginning)

# T D - Commands - monitoring

- Additional Monitoring Facilities:
  - Display System Activity IUI Dialog
    - `"SYSTEM (CPUs:  2 /  0 )"`
      reflects number of CPUs active / quiesced
    - `"CPU : 127%"`
      reflects TOTAL CPU time for all processors
    - PF2 gives display of balanced partitions (if any)
    - PF5 gives display of dynamic partition classes
    - PF6 gives graphical CPU summary

# T D - Commands - monitoring

- Additional Monitoring Facilities:
  - Graphical monitoring via OS/2 or Windows and VSE/ESA Distributed Workstation Feature
  - VSE Workdesk folder, VSE CPU Activity icon monitors CPU activity via
    - Bar chart (current snapshot)
    - History chart (up to 100 recent intervals)
    - APPC or file transfer connection

# T D - Performance Hints

- Using the Turbo Dispatcher Efficiently
  - T D Efficiency depends on number and type of work units available
  - More parallel work units
  - Lower non-parallel share (NPS)
  - T D can exploit more CPUs

# T D - Performance Hints...

- Multiprocessor Eligibility
  - Workload CPU utilization per partition
    - Peak period (hour)
    - Display System Activity, Job Accounting, or Performance Monitor data
  - Adjust for changes in requirements
    - Expected (or intended) growth
    - Release transition requirements
    - Additional T D overhead
    - Multiprocessor overhead

# T D - Performance Hints...

- Multiprocessor Eligibility...
  - Adjust for expected efficiency improvements due to use of new technologies (data-in-memory), splitting large CICS or batch partition workloads.
  - Select that n-way processor which provides sufficient processing power on a single CPU for
    - The largest partition workload
    - The total non-parallel workload
    - Enough total CPUs to handle total concurrent workload requirements

# T D - Performance Hints...

- Maximum Number of Exploitable CPUs
  - Determine the non-parallel share of your workload
    - Run with Turbo Dispatcher
    - Use QUERY TD output
    - Use VSE CPU Activity from VSE Workdesk
    - (Can be done on uni-processor)
  - For earlier releases (VSE/ESA V1...), estimate...

# T D - Performance Hints...

- Maximum Number of Exploitable CPUs...
  - For earlier releases (VSE/ESA V1...), estimate using the chart below:

| Type of VSE/ESA Workload | Non-Parallel Share (NPS) |
|---|---|
| CICS (With data-in-memory) | About 0.3 |
| CICS (Without data-in-memory) | About 0.4 |
| Batch (With heavy I/O activity) | About 0.5 |

# T D - Performance Hints...

- Maximum Number of Exploitable CPUs...
  - Calculate

    **`number of CPUs = 0.9 / non-parallel share`**

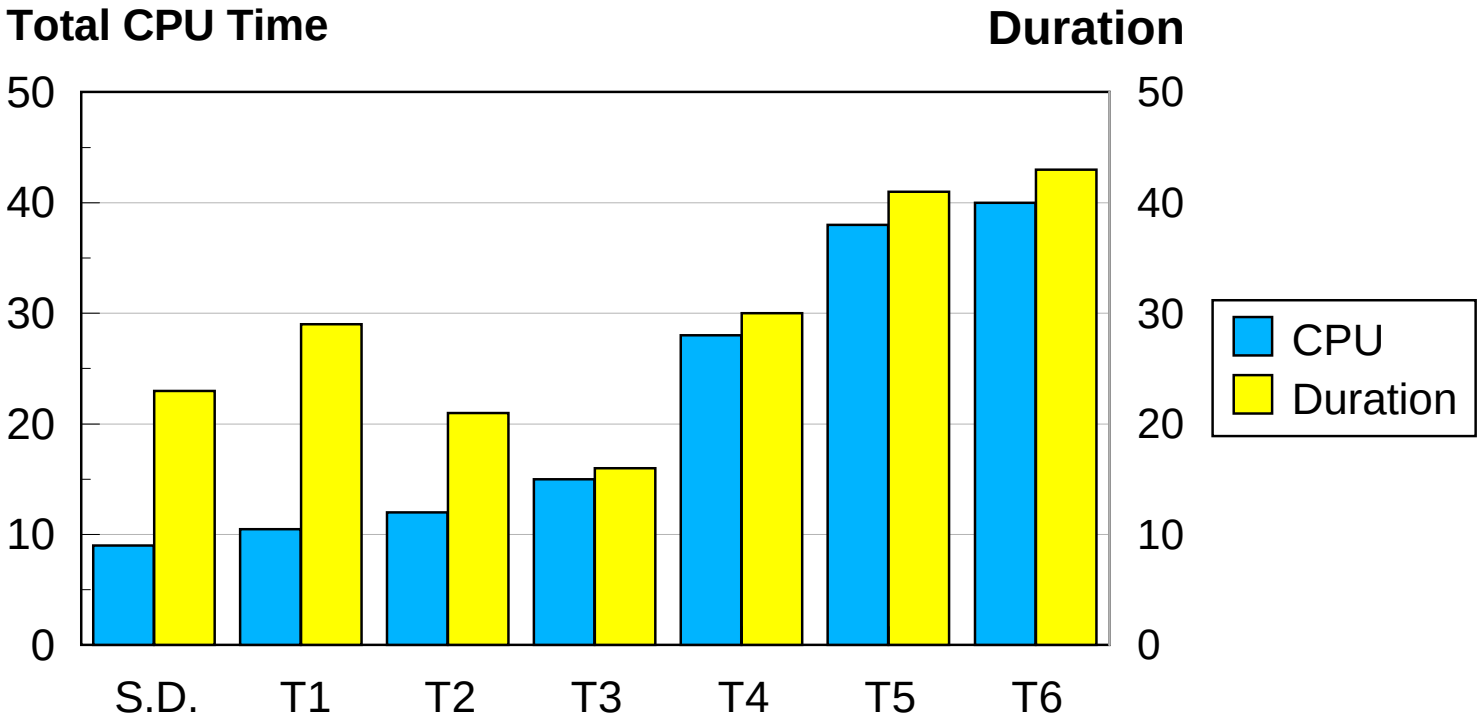
    (0.9 factor considers delays queueing for non-parallel state)
  - Or use this table:

| Non-Parallel Share (NPS) | 0.25 | 0.30 | 0.35 | 0.40 | 0.45 | 0.50 | 0.55 |
|---|---|---|---|---|---|---|---|
| Maximum Number of Fully Exploitable CPUs | 3.6 | 3.0 | 2.6 | 2.2 | 2.0 | 1.8 | 1.6 |

  - Extra (unexploitable) CPU power adds overhead -- use 3 engines instead of 3.6, for example.

# T D - Performance Hints...

- Maximum Number of Exploitable CPUs...




NPS~0.25 -- 3.6 Engines can be exploited

"SD"=Std. Disp;  "Tn"=Turbo with "n" engines

# T D - Performance Hints...

- Selecting a Processor
  - T D exploits up to 4-way processors with known workloads
  - T D can tolerate up to 10-way processors
  - Consider 2 to 4 CPUs depending on above
  - Under VM or PR/SM, this is guest or LPAR, perhaps on processor with more CPUs.
  - Start with lower number of CPUs than indicated, consider trying an additional engine.
  - A guest/LPAR with fewer CPUs than processor incurs less overhead than if all CPUs used

# T D - Performance Hints...

- Partition Considerations
  - **AT MOST** one partition can use one CPU
  - Other partitions may cause that partition to wait for the non-parallel state.
  - A single CPU's power should exceed a single partition's needs to cover this effect.
  - The more concurrently active partitions exist, the more work units are available for dispatch
  - The lower the non-parallel share of work units, the smaller the wait for the non-parallel state.

# T D - Performance Hints...

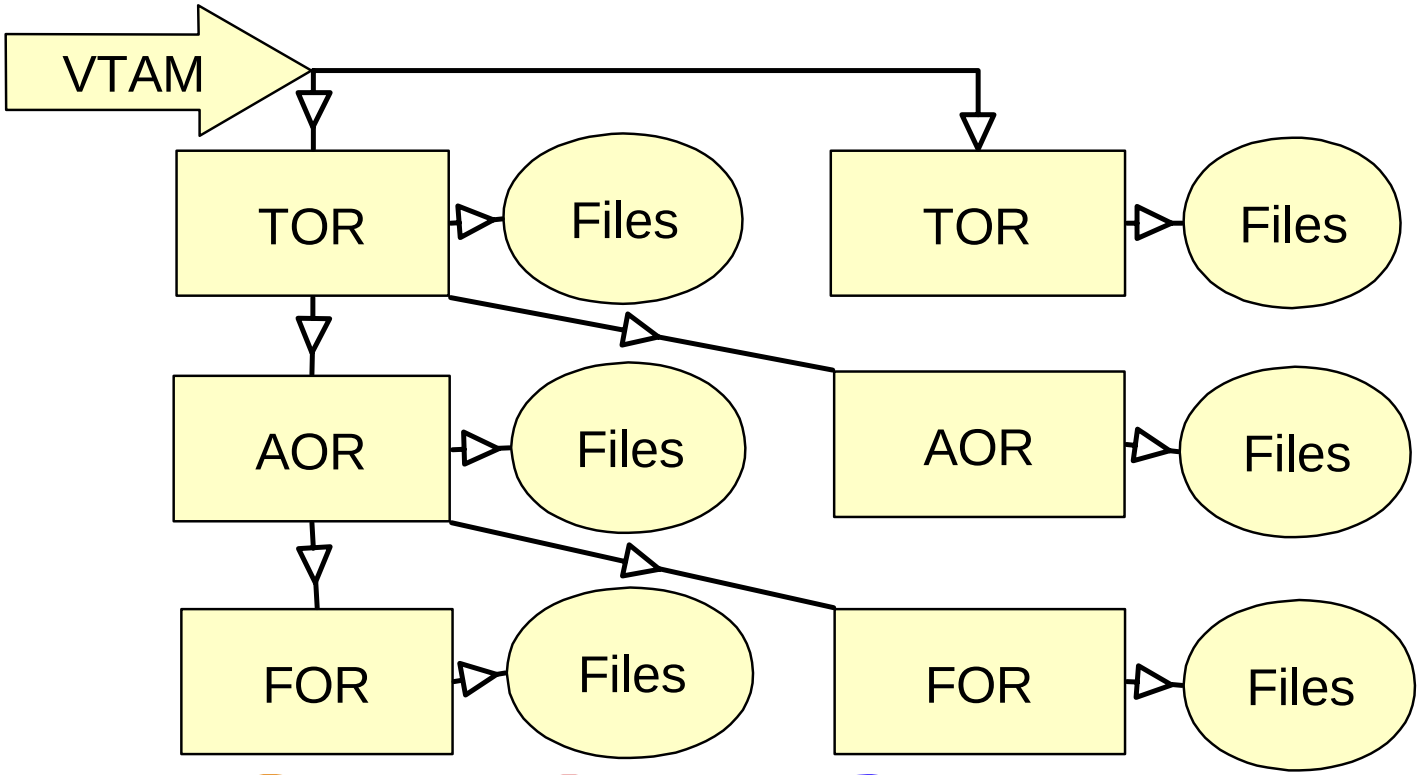
- Partition Considerations
  - **Large** batch or on-line partitions may benefit from splitting them
  - Application design factors are different for batch with Turbo Dispatcher to exploit many CPUs.
  - Concurrency is REQUIRED
    - Multiple steps don't help
    - You need multiple jobs
    - Job scheduling schemes must change

# T D - Performance Hints...

- Partition Considerations
  - CICS work can be split into multiple partitions
    - TOR/AOR/FOR or combinations are used
    - CICS MRO is used for communication
    - Minimize MRO overhead by
      - Using independent CICS partitions
      - Using CICS Transaction Routing to distribute
      - Arrange applications and files to minimize
      - Exploit CICS Shared Data Tables

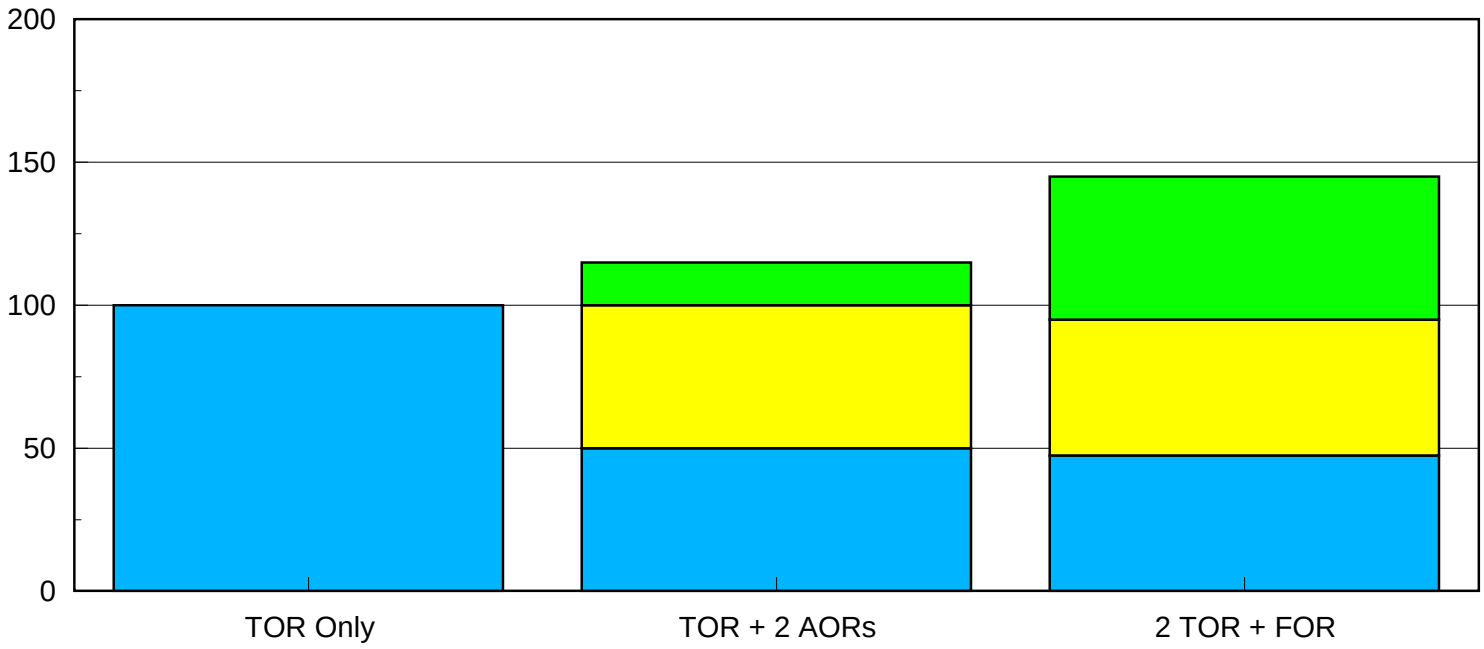
# T D - Performance Hints...

- Partition Considerations
  - CICS work can be split into multiple partitions...



# T D - Performance Hints...

- Partition Considerations
  - CICS work can be split into multiple partitions...
  - MRO cost depends on configuration



# T D - Performance Hints...

- Uni-processor considerations
  - The good news is
    - None of this directly applies, but
    - The tuning done to reduce I/O and overhead on a uni-processor reduces NP share (which has no effect on a uni-processor) and T D overhead -- positioning the system for multiprocessor exploitation.
    - Information can be gathered to show how the system will perform on a multiprocessor.

# T D - Summary

- T D standard in VSE/ESA 2.4
  - Function needed for CICS Transaction Server
- T D provides added new functions
  - Balance of dynamic partitions rather than classes
  - Share balancing
  - Multiprocessor support
    - Options for higher throughput at lower cost
    - Single image VSE systems with multiple CPUs
    - Reduced need for system sharing facilities

# T D - Summary...

- T D has additional costs
  - CPU overhead
    - low utilization effect
  - For multiprocessors, additional considerations:
    - Non-parallel share effects
    - Workload characteristics
    - Single engine speed vs. Total processor power