The Swami's Performance Methodology Ideas

Dan Janda
The Swami of VSAM

The Swami of
VSE/VSAM

Dan Janda
VSE, VSAM and CICS
Performance Consultant
RR 2 Box 49E
Hamlin Road
Montrose, PA 18801-9624

(570) 934-2862
theswami@epix.net
http://business.epix.net/~theswami

World Alliance of VM,VSE and Linux
Chattanooga, Tennessee
April 2006

The Jargon of VSAM, File Systems and Performance

- The topic of performance can mean many things to many people
  - Speed -- how fast is the CPU -- often referred to as MIPS
    - MIPS -- "Meaningless Indicator of Processor Speed" unless the architecture and workload used for measurement are defined
  - Power -- how much work can be done by this CPU in a unit of time
    - If the CPU has more engines, its throughput capacity will be higher than a CPU with fewer engines of the same speed
  - Throughput -- number of jobs, transactions, or other units of work per unit of time
    - This is often not repeatable on a detail level because of random variation of workload factors
  - An approximation: Engine Speed x Number of Engines = Power
    - As you add engines, the effective speed of each engine is less

The Jargon of VSAM, File Systems and Performance

- The topic of performance can mean many things to many people...
  - Time terms:
    - Elapsed Time -- the difference in "wall clock" times between start and end of a process, job, transaction, or other unit of work -- also Response Time
      - The sum of the Elapsed Times of a group of concurrent jobs exceeds the elapsed time of the group
    - CPU Time -- the amount of time the CPU was busy actively processing a unit of work -- in VSE, measured by a system facility called the CPU Timer
      - On multi-processor CPUs, the amount of CPU time available per unit of time is equal to the number of engines times the unit of time.
      - On a 3-engine CPU, 180 seconds of CPU time is available each minute
    - Wait Time -- the amount of time the CPU was not busy actively processing a unit of work
      - Total wait time for a job may exceed the elapsed time
    - Overhead Time -- Time the CPU was busy but its activity was not directly attributable to one or another unit of work
      - Paging, initial I/O interrupt handling, ...

The Jargon of VSAM, File Systems and Performance

- The topic of performance can mean many things to many people...
  - Input and Output (I/O) performance terms...
    - I/O rate -- number of I/O operations per unit of time
    - The following items account for the elapsed time of a single I/O operation on a DASD (or disk device) -- I/O Response Time
      - I/O queueing time -- time spent by a process waiting its turn to use an I/O device -- queueing within the operating system, generally
      - Pend time -- queueing time within the I/O subsystem
      - Seek time -- time during which a disk device arm is in motion
      - Rotational Delay time -- time during which a disk rotates to the position where the desired data is to be found
      - Transfer time -- time during which data is moved from I/O device to CPU storage
    - In addition, CPU time is used during I/O activity to prepare channel programs, translate them to reflect real storage addresses, and then to process the interrupt indicating the completion of the I/O operation

Performance Basics

- What is performance?
- How can you define performance?
- How does your boss (or your boss's boss) define performance?
- How can you measure performance?

- Let's understand some of the pro's and con's of various alternatives...

Performance Basics

- **Performance Views**
  - **Overall System View -- throughput perspective**
    - **Jobs per hour, shift, day**
    - **Average utilization**
    - **I/O rate**
    - **. . .**
  - **Subsystem View -- throughput or speed perspective**
    - **Transactions per second**
    - **I/O rate per device**
    - **. . .**
  - **Job step (or transaction) view -- speed or throughput perspective**
    - **Elapsed (or response) time**
    - **I/Os (overall or by device)**
    - **CPU time, wait time**
    - **. . .**

Performance Basics

- **All CPUs wait at the same speed**
  - **It does not matter to the critical job why it is waiting**
    - **It could be waiting because another job (of higher priority) is running**
    - **It could be waiting because it has requested an I/O operation**
    - **In fact, the CPU could be executing instructions on behalf of the critical job which could have been avoided by:**
      - **Better program design**
      - **Better program coding (or optimization)**
      - **and the processing required by the critical job is not being done**
    - **I/O operations (and the CPU time to manage them) that could be avoided**

Performance Basics

- All CPUs wait at the same speed
  - They Wait for work
    - The CPU may be idle
    - The CPU may be processing lower priority work
  - They Wait for I/O
    - During I/O operations
    - The CPU may be processing lower priority work
- In each of these cases, the processor is not processing this job
- Our role as performance people is to
  - Reduce the amount of this wait
  - Reduce the amount of CPU resource used to accomplish a task
  - Free system resources for use by other tasks

Performance Basics

- **All CPUs wait at the same speed**
  - **All I/O operations require CPU time for their management**
    - **Depends on type of I/O, environment**

Type of Operation	Native VSE	Typical VM/VSE	Optimum VM/VSE
Tape, Real Printer	1.0-2.0K	2.0-4.0K	1.5-3.0K
CKD DASD	10-20K	20-40K	15-30K
ECKD DASD	8-16K	16-32K	12-24K

1000 long DASD I/Os / second could be 40 Mips

- **The Swami's educated estimates -- your mileage will vary**

Performance Basics

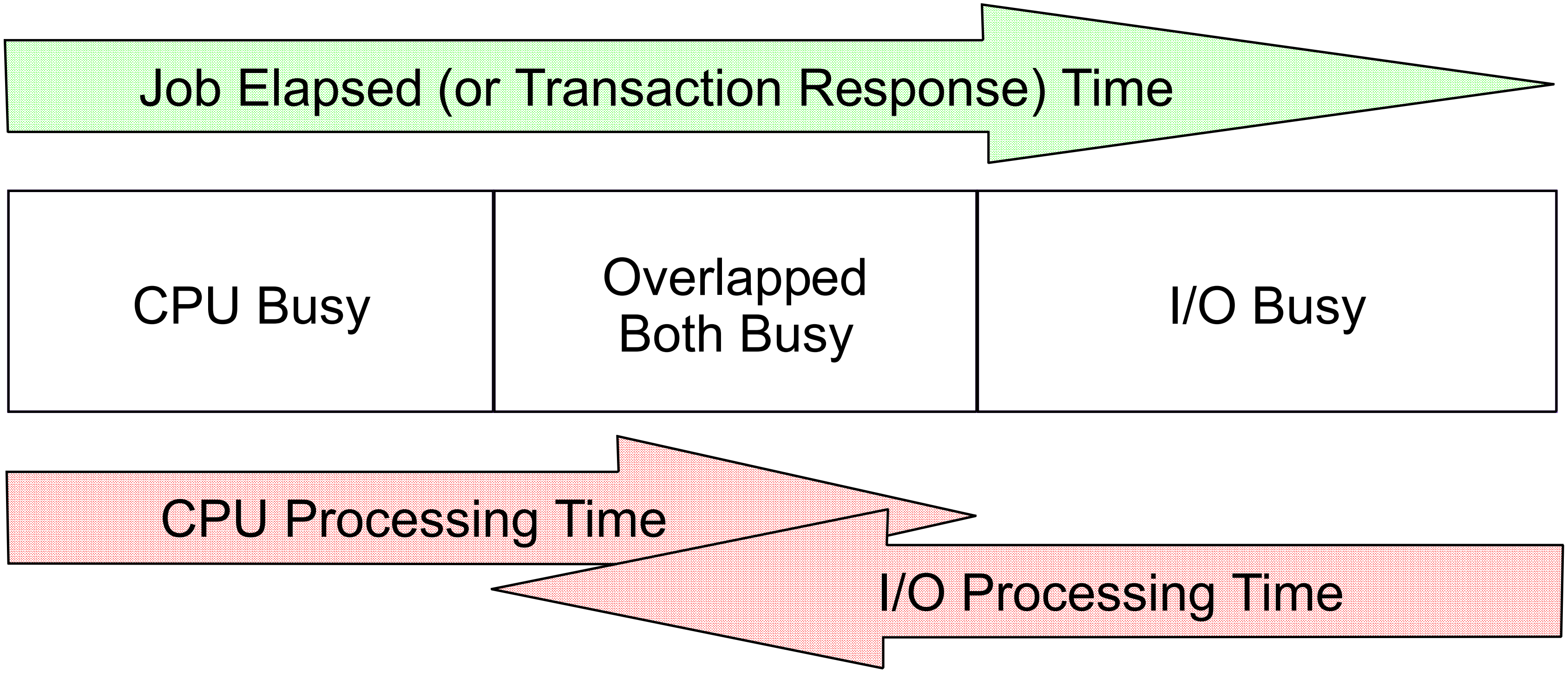
- **CPU Dispatching Priority vs. Job Priority vs. Job Importance**
  - **Job Importance...**
    - **Printing paychecks is more important than playing solitaire**
  - **Job Priority...**
    - **POWER job scheduling priority controls sequence jobs start within POWER's work classes**
  - **CPU Dispatching Priority...**
    - **The order in which the operating system dispatcher will search for work to be done among those tasks "ready to run"**
- **VSE permits dispatcher priority to be dynamically changed (balanced) among workloads based on their CPU usage during a measurement interval**
  - **The VSE balancing measurement interval is set by the IPL MSECS parameter**
    - **The system default (about 1 second) seems reasonable in most cases. Try smaller value for >100 Mips processors and short job steps**

Performance Basics

- A CPU can only be in one of two states:
  - Running
  - Waiting, divided into
    - Waiting for work -- there is nothing for it to do -- it is idle
    - Waiting for I/O -- one or more I/O activities have been started, but no work can continue until one of them completes
- A job can only be in one of four states:
  - CPU Running with no I/O operations active for this job
  - CPU Running together with I/O operations for this job active
  - CPU Waiting while I/O operations for this job are active
    - Lower priority job(s) may be running on the CPU
  - CPU Waiting with no I/O operations for this job active
    - Higher priority job(s) are running on the CPU
    - Wait for operator, scheduling holds, etc.
      - These may be (indirectly) other jobs' I/O causing the delay

Performance Basics

- A simple job example shows some basic concepts:
  - Job reads records from file
  - Job computes result using data from those records



Job Elapsed (or Transaction Response) Time

CPU Busy

Overlapped
Both Busy

I/O Busy

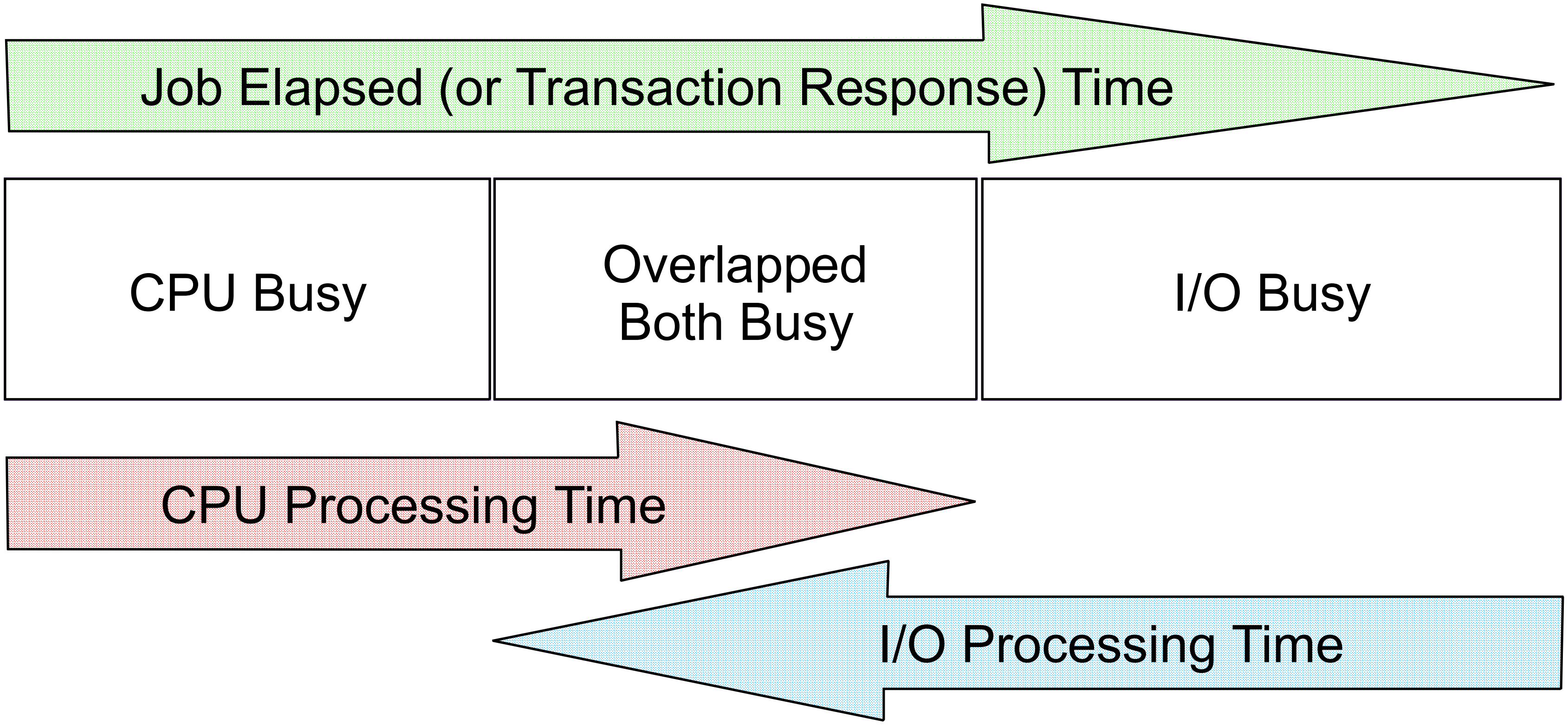
CPU Processing Time

I/O Processing Time

This description of a workload is called its "profile"

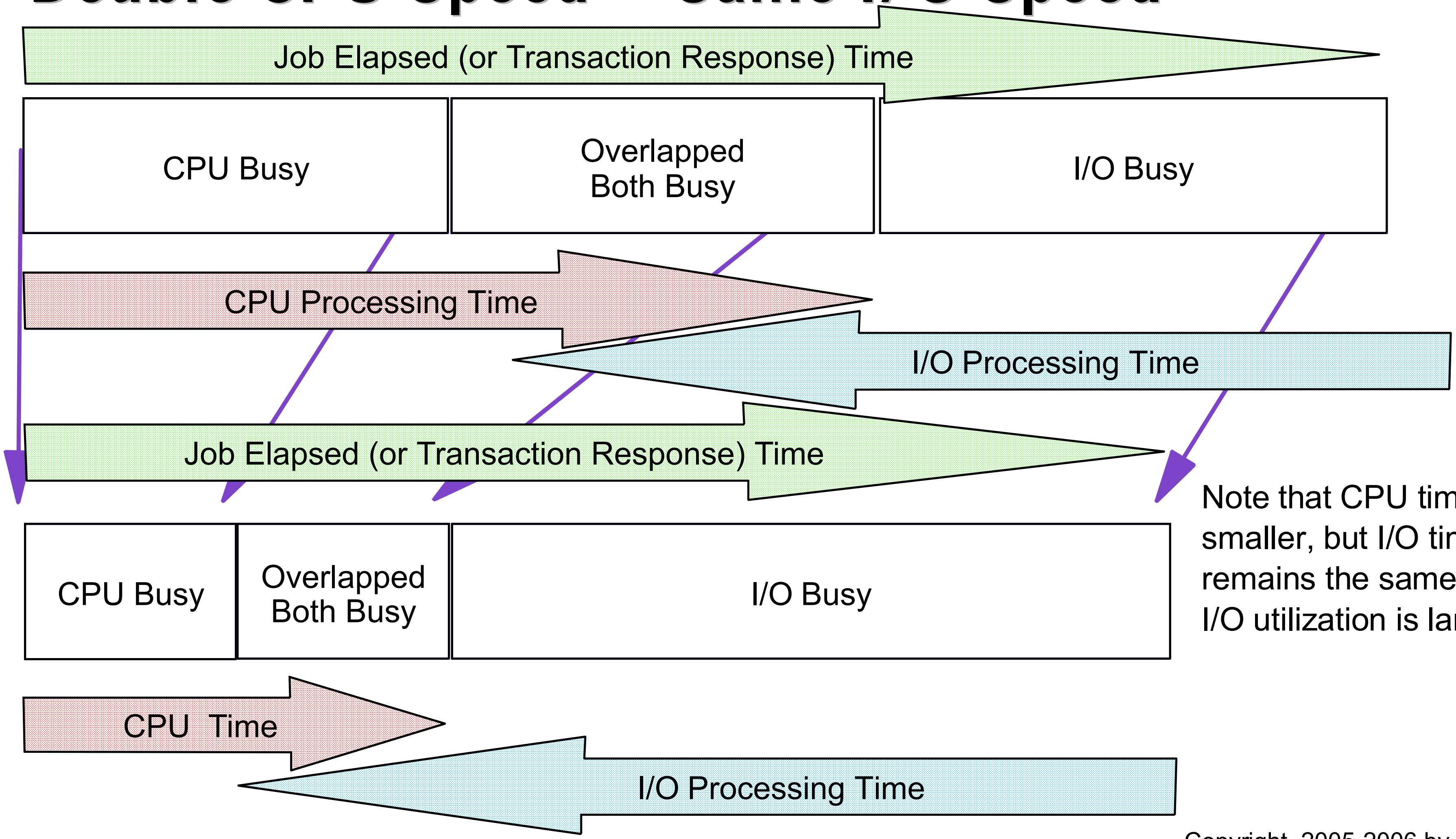
Performance Basics

- System changes' effects can be seen:
  - CPU changes affect only CPU Busy and Both Busy segments
  - I/O changes affect only I/O Busy and Both Busy segments



Performance Basics

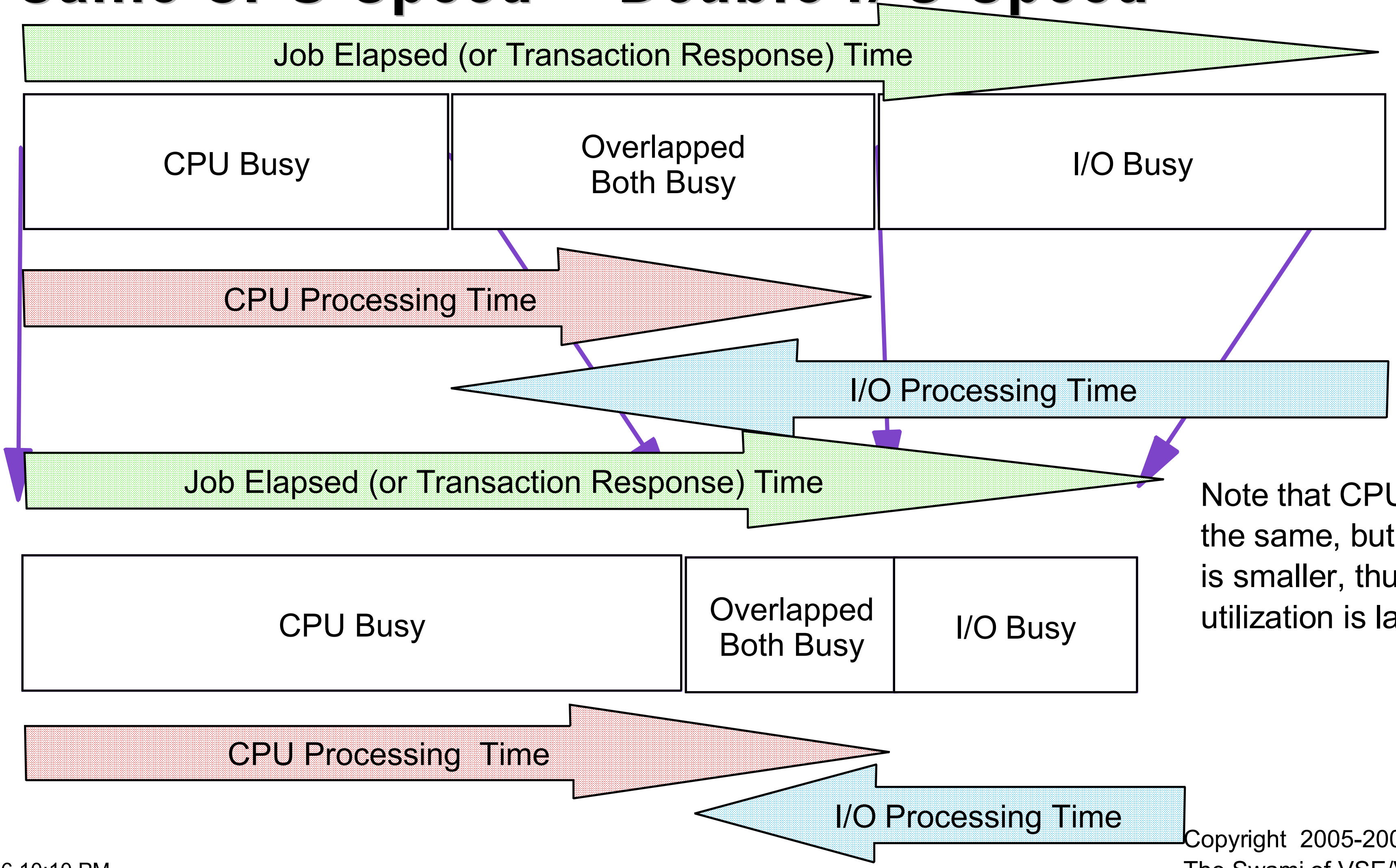
Double CPU Speed -- Same I/O Speed



Note that CPU time is smaller, but I/O time remains the same, thus I/O utilization is larger

Performance Basics

- **Same CPU Speed -- Double I/O Speed**



Job Elapsed (or Transaction Response) Time

CPU Busy

Overlapped
Both Busy

I/O Busy

CPU Processing Time

I/O Processing Time

Job Elapsed (or Transaction Response) Time

CPU Busy

Overlapped
Both Busy

I/O Busy

CPU Processing Time

I/O Processing Time

Note that CPU time is the same, but I/O time is smaller, thus CPU utilization is larger

Performance Basics

- This technique is called Profile Conversion...
  - A simple approach to predicting the effect of system changes on system performance
  - Rigorous mathematical basis, but only simple math is needed to use the process
  - Gathering data for profile conversion is not trivial
    - Tools to measure Elapsed time and CPU time exist
    - Tools to measure I/O time and Overlapped time do not exist (as far as I know -- but see next page)
    - IBM's VSE/PT produced data for this purpose, but it is no longer available nor operational
  - Reasonable estimates can be made in many cases

Performance Basics

- Profile Conversion...
  - Gathering data for profile conversion is not trivial
    - Reasonable estimates can be made in some cases
    - VSE's SIR command can be very useful

```
SIR
SIR ?                             Displays SIR commands
SIR RESET                         Resets SIR counts/totals
SIR SMF[={ON|OFF}][,VSE][,cuu]    Subsystem Measurement Facility
```

Performance Basics

- Profile Conversion...
  - Reasonable estimates can be made in some cases
  - Job Accounting gives us CPU time accurately
  - Job Accounting gives us number of I/Os by device
  - EXPLORE, TMON, OMEGAMON can give us average I/O time for devices during the time in question
  - Total I/O time for a device is just product of the **Number of I/Os** and **Average I/O Time** for device
  - Sounds like something a spreadsheet could do!
    - Calculators, pencils, and similar tools can too!

CPU Busy

Overlapped
Both Busy

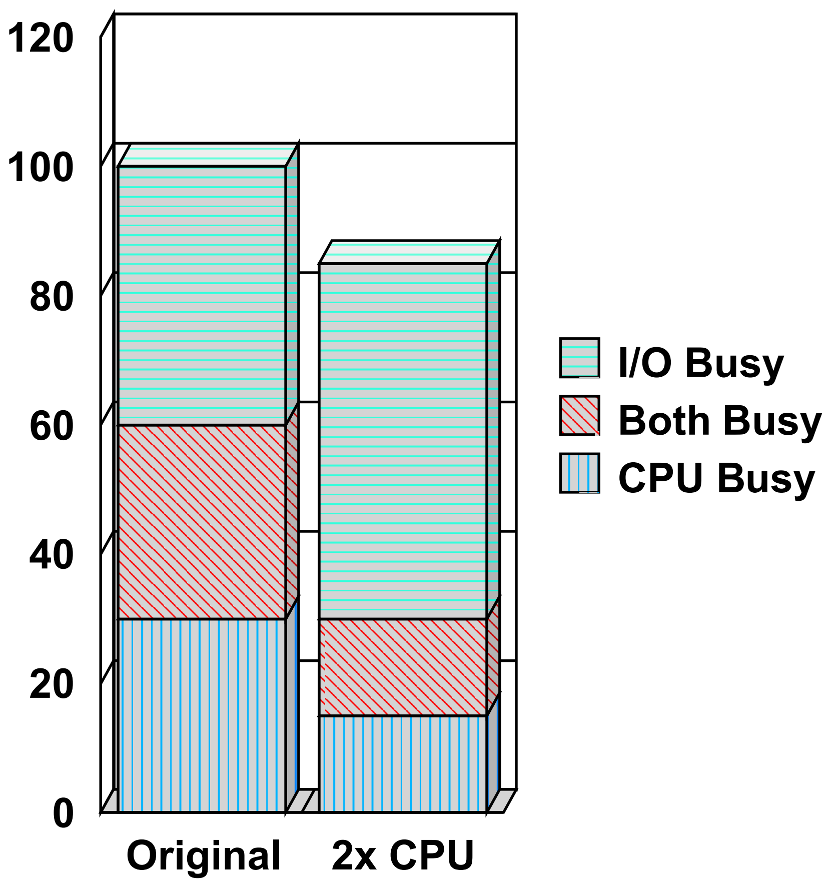
I/O Busy

Performance Basics

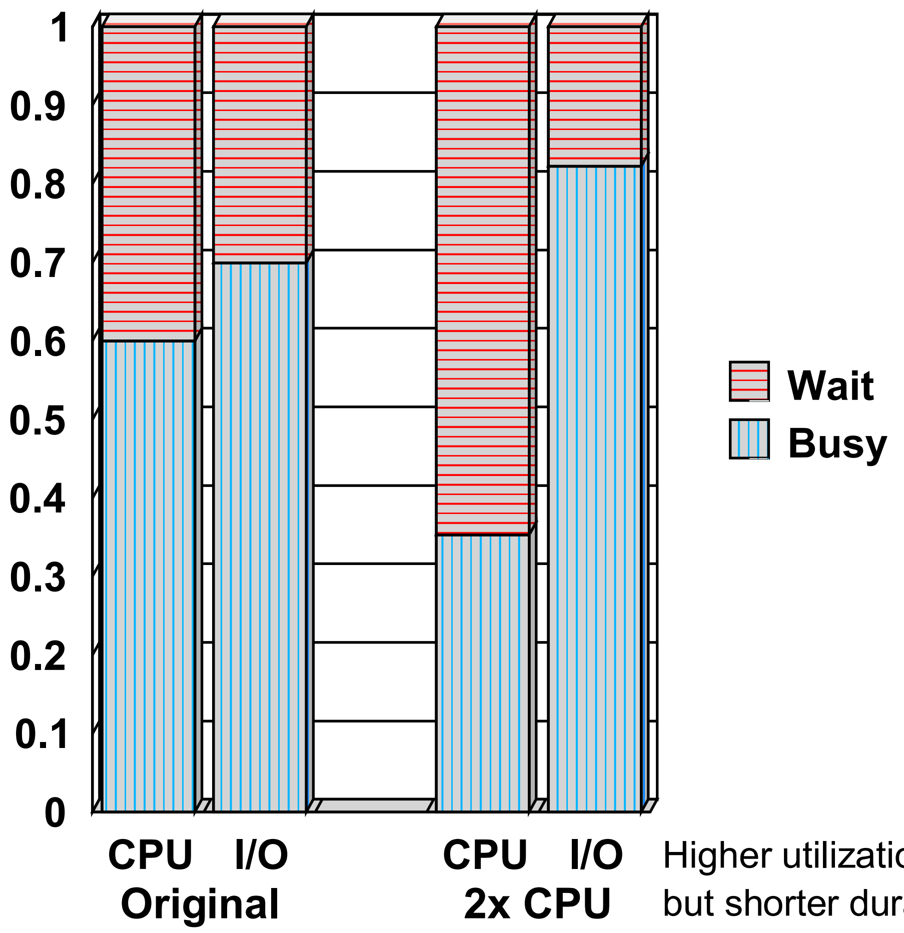
- **Resource Utilization issues**
  - **Increase the speed of a component (CPU or some I/O device(s))**
  - **The system runs the workload faster (less elapsed time)**
  - **Wait on that component is now less**
  - **Utilization of other component(s) increases during the (now shorter) duration of the workload**
  - **Another component will immediately become the key bottleneck for that workload**

Performance Basics

Resource Busy Time



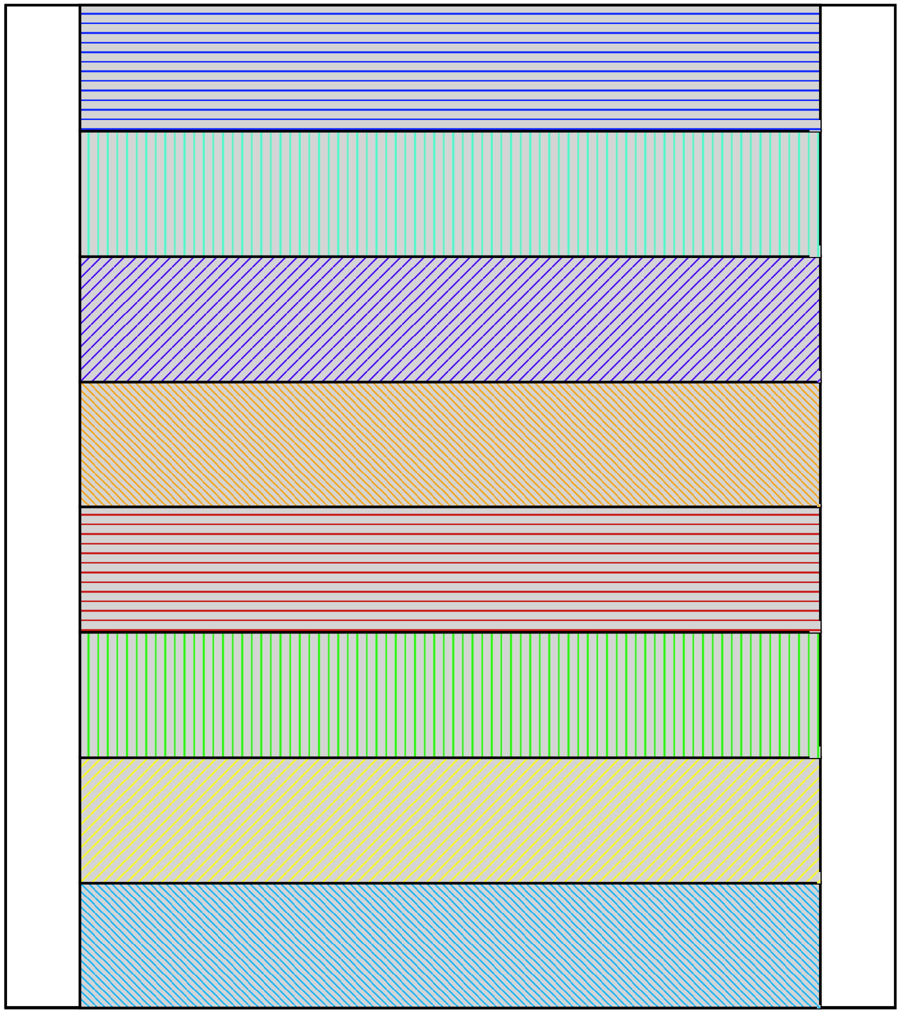
Resource Utilization Percentage



Higher utilization
but shorter duration

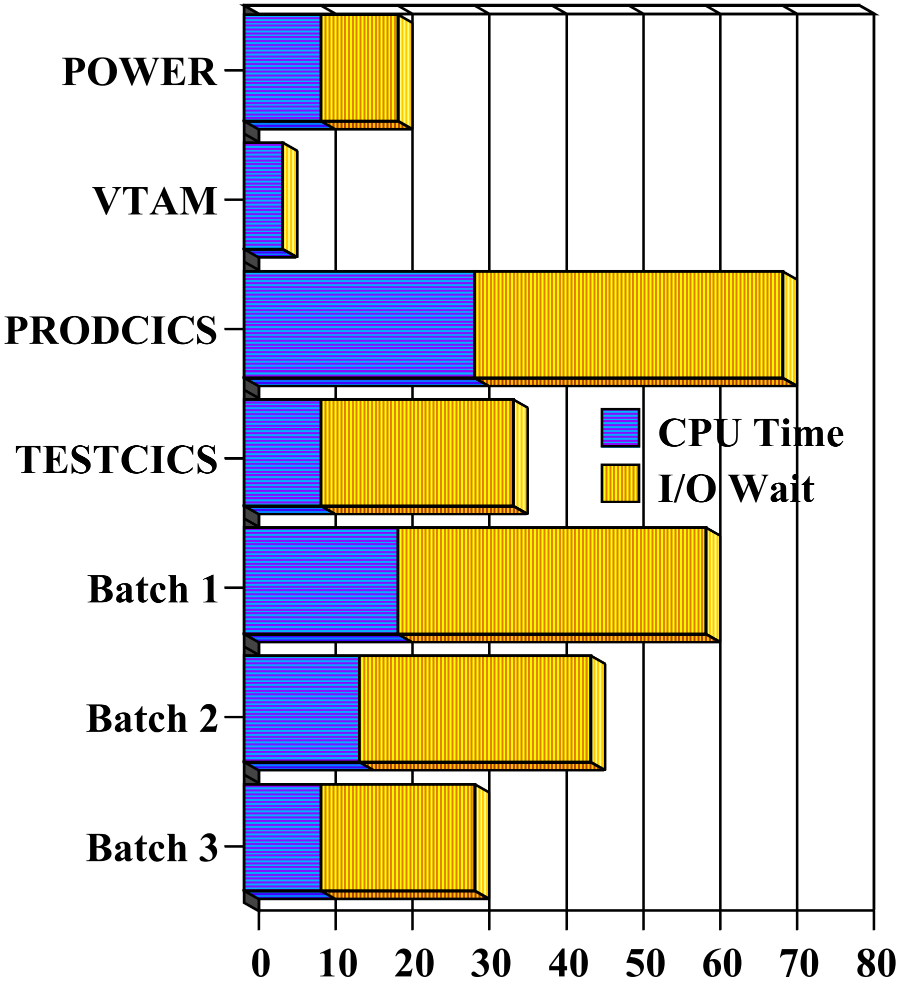
Performance Basics

- **VSE uses a Preempt/Resume Priority Dispatch Algorithm**
  - **The highest priority task that is ready-to-run is dispatched**
  - **When that task must wait, the next highest priority job is dispatched**
  - **When any event completes, the dispatcher suspends the running task and re-evaluates the status of all tasks**
  - **If no task is ready-to-run, then CPU waits for work**



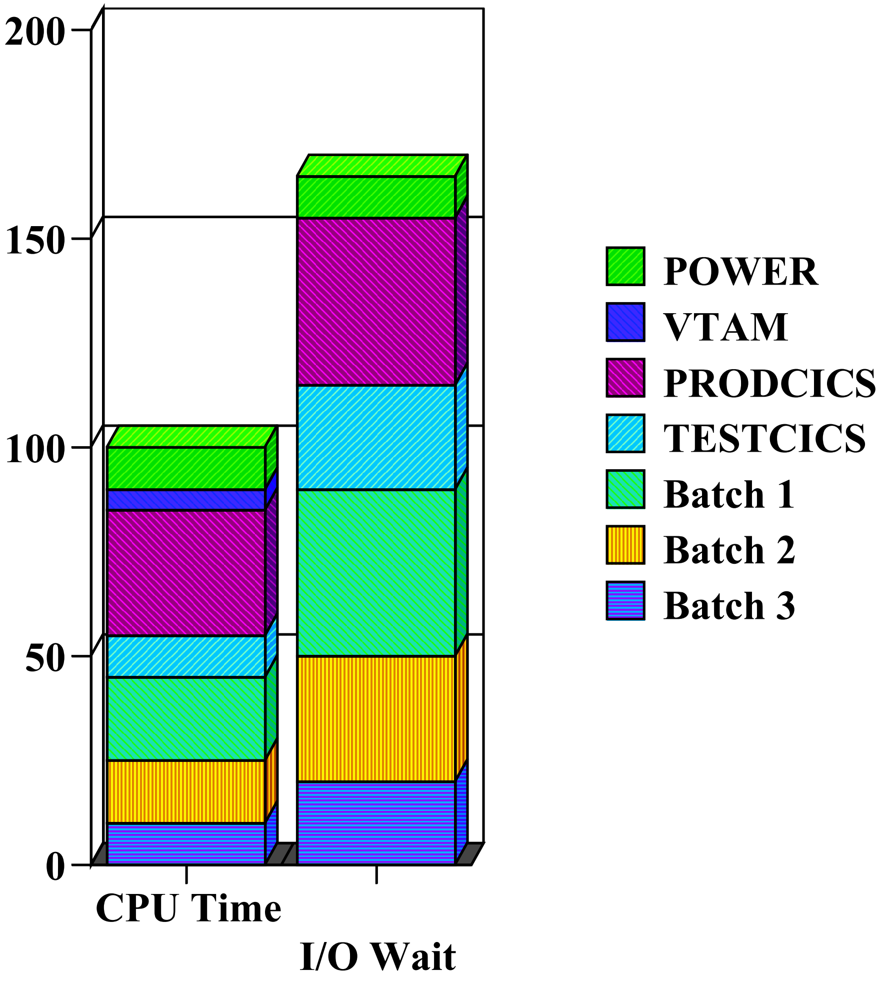
Performance Basics

- CPU is used by high priority task
- Lower priority tasks get what's left over
- If no CPU time is left, the lowest priority tasks get no CPU time
- "Effective CPU Speed" is the fraction of the total CPU speed left after higher priority tasks have used what they will
  - In this case, Batch 3 "feels" a CPU only 1/10th of the full power
  - Tuning a higher priority task to reduce its I/O wait will reduce Effective CPU Speed for lower priority tasks



Performance Basics

- CPU is used by high priority task
- Lower priority tasks get what's left over
- If no CPU time is left, the lowest priority tasks get no CPU time
- "Effective CPU Speed" is the fraction of the total CPU speed left after higher priority tasks have used what they will
    - In this case, Batch 3 "feels" a CPU only 1/10th of the full power
    - Tuning a higher priority task to reduce its I/O wait will reduce Effective CPU Speed for lower priority tasks while the higher priority task is running



Performance Basics

- CPU is used by high priority task
- Lower priority tasks get what's left over
- VSE/ESA's Turbo Dispatcher partition balancing
  - All partitions in a dynamic class are balanced
  - PRTY command can balance static partitions and partitions in dynamic classes
    - PRTY F1,F3,F2,G=F4=F5=BG indicates that dynamic partitions in class G and static partitions F4, F5, and BG form a balanced group
    - Only one balanced group can exist
    - PRTY SHARE,G=100,F4=200,F5=200,BG=50

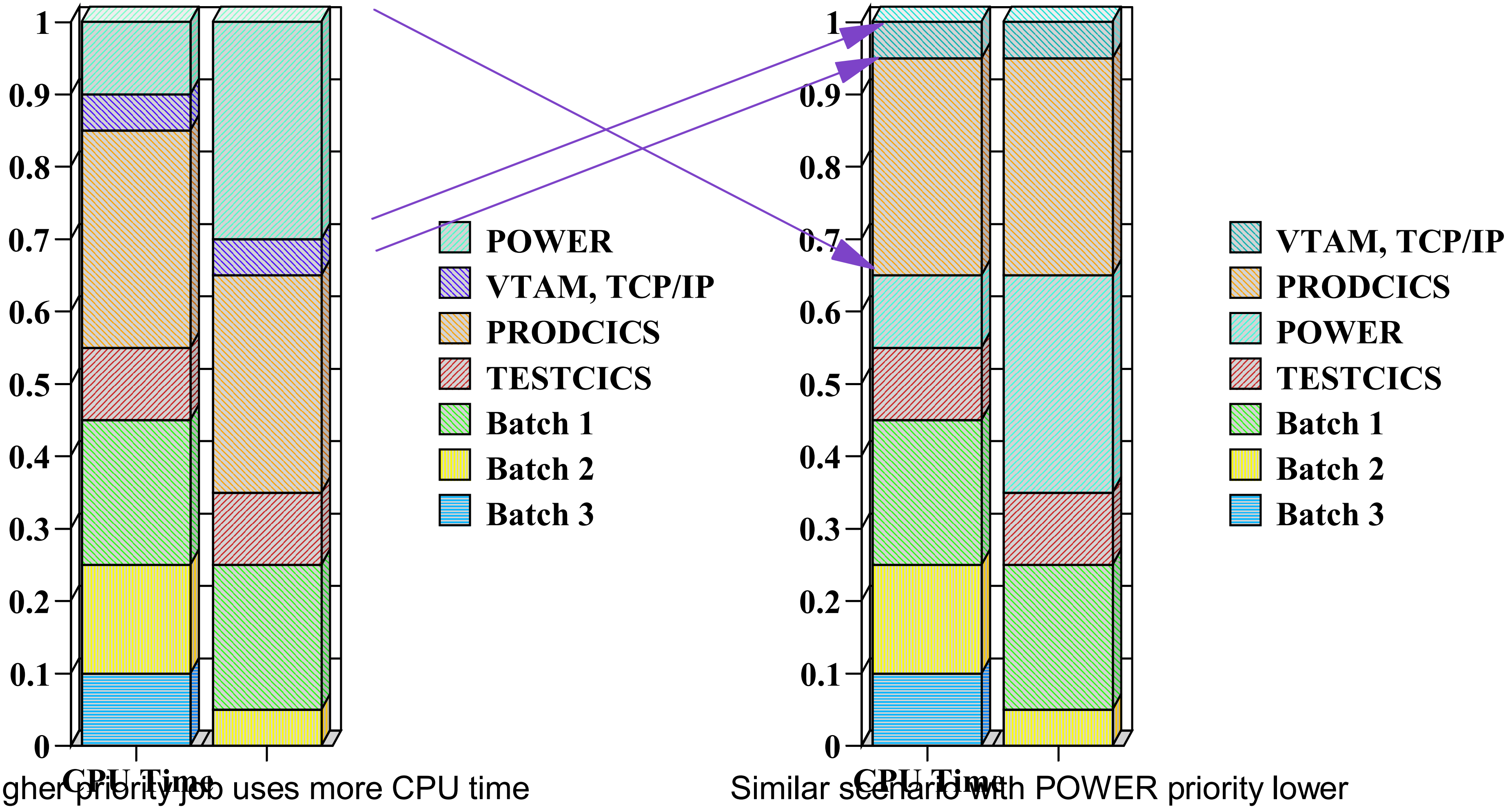
Performance Basics

- CPU is used by high priority task
- Lower priority tasks get what's left over
- VSE/ESA's Turbo Dispatcher partition balancing
  - PRTY command can balance partitions and dynamic classes
    - PRTY SHARE,G=100,F4=200,F5=200,BG=50
      - Each class G partition will receive an equal share, and the other partitions in the balanced group will receive an equal, greater, or lesser share based on the relative values specified
        - F4 and F5's shares are twice the share of the dynamic partitions
        - BG's share is half the share of the dynamic partitions
      - If a partition uses up its share of CPU resources, other partitions in the balanced group will be dispatched
      - Partitions which have used their full share are eligible to use more resources if any are available after other partitions in the group have received their share or are waiting

Performance Basics

- Some special cases:
  - POWER provides services primarily for batch
    - Batch jobs with large print components will cause POWER CPU utilization to increase
      - I call this "reflected batch work"
    - But much of it is processed at POWER's priority rather than the batch priority
    - This can impact VTAM, CICS, and data base services
  - VTAM provides services for CICS, POWER, and TCP/IP...
    - Its CPU use is generally charged to the using partition
    - It is difficult to see VTAM's CPU consumption directly
      - In test environments with scripted workloads, repeatable tests can be done and are the basis of VTAM tuning advice

Performance Basics



Higher priority job uses more CPU time
and impacts all lower priority jobs including on-line

Similar scenario with POWER priority lower
so on-line is not impacted -- only batch

Performance -- not quite basics

- So, what can we do about it?
  - Measure
    - Data
      - Elapsed time, CPU time, I/O time; I/O counts
    - Performance Monitors
      - Explore, TMON, others
    - Job Accounting
      - CA-JARS, others
      - $JOBACCT, POWER accounting routines
    - Other sources -- Turbo Dispatcher
      - QUERY TD[,INTERNAL]
        - Dependent upon control of workload
      - SYSDEF TD,RESETCNT
    - VM, hardware data

Performance -- not quite basics

- Some real-world considerations for measurements
  - Multiprogramming effects
    - Impact of other workloads
    - Repeatability -- key factor
      - CPU time
        - Repeatable
      - I/O counts
        - Repeatable
      - Elapsed time
        - Not repeatable
      - I/O time
        - Not repeatable

Performance -- not quite basics

- SKJOBACC routine in ICCF Library 59
  - Captures VSE Job Accounting data at step end
    - Elapsed time
    - CPU time
    - Overhead time
    - I/O counts by device
  - Display results as a printed page on SYSLST
  - You have source code, so you can modify if desired
    - Simplest:
      - Output on another virtual printer
    - More complex:
      - Output on console
      - Change format, etc...
      - Selective output
        - (e.g. only for jobs named "TUNE....")

Performance -- not quite basics

- **SKJOBACC Metrics**
  - **Elapsed time**
    - **Interesting, but not useful unless complete workload is repeatable**
  - **CPU time**
    - **Very useful, repeatable, and can compare tuning and environmental changes with good precision**
  - **Overhead time**
    - **CPU time not identifiable for a specific task, apportioned among all active tasks on a pro-rata basis**
  - **I/O counts by device**
    - **Specific activity by job step to each device used**
  - **Others: Paging, POWER functions...**

Performance -- not quite basics

- What can I do with this (SKJOBACC or similar) data?
  - I can tune a job step to
    - improve its performance
    - minimize its impact on other concurrent jobs
  - I can make tuning decisions for this job step
    - based on solid evidence
    - even when other jobs are running concurrently.

Data near programs?

Method	Elapsed Time microseconds	CPU Time microseconds
COBOL Working Storage	1	1
CICS Data Table	10	10
VSAM LSR hit	25	25
VSE Virtual Disk	100	100
VM Virtual Disk	250	250
Well-cached real disk	1000	800
Un-cached disk	20000	800

Contacting the Presenter

- For more information...
  - You can contact the Swami by e-mail

    theswami@epix.net

  - He's building a web site about VSE/VSAM issues

    http://business.epix.net/~theswami

  - Downloadable ".PDF" files of the handout for this presentation can be found by following the links on that web page.

  - His knowledge and experience can help you, too!