

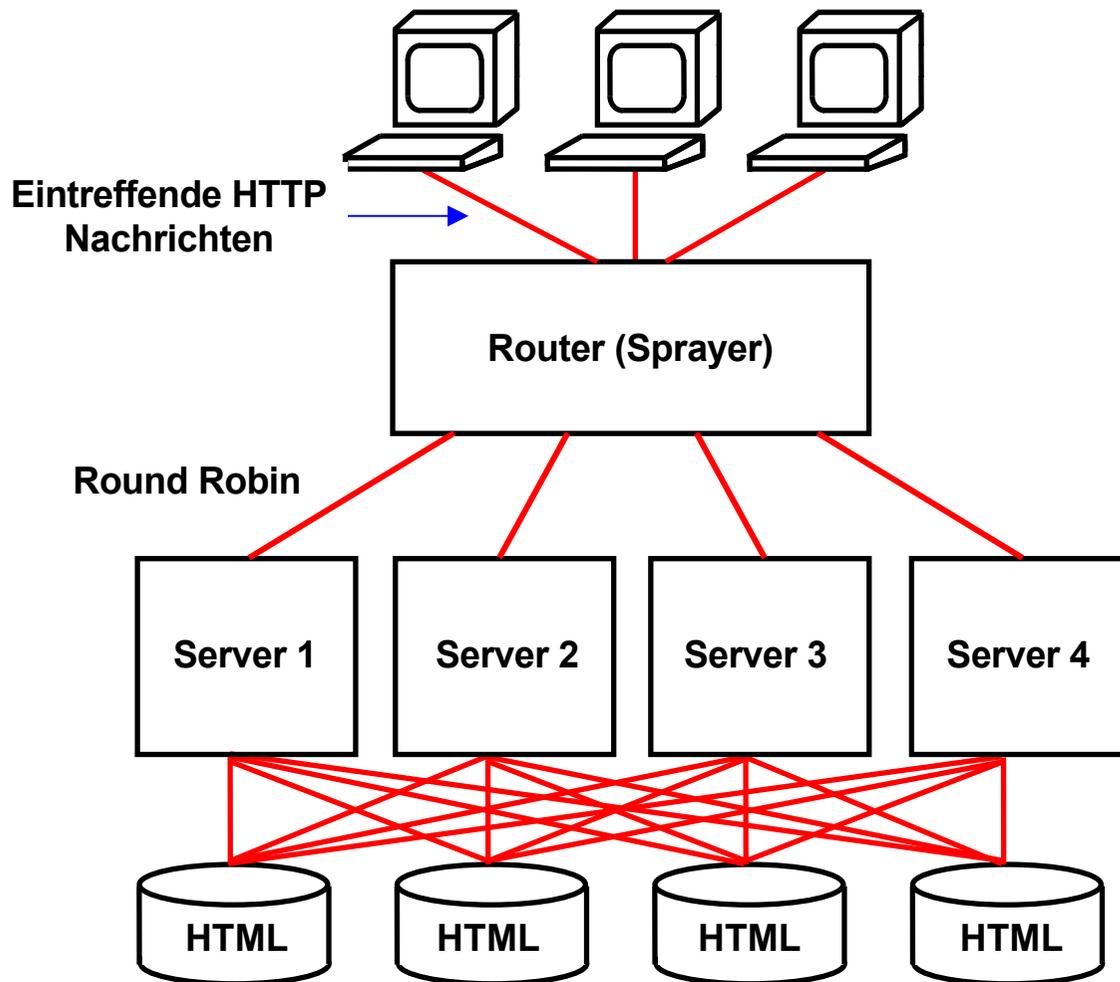
# Mainframe Internet Integration

Prof. Dr. Martin Bogdan  
Prof. Dr.-Ing. Wilhelm G. Spruth

SS2013

Work Load Management Teil 1

Work Load Manager Komponenten



## Work Load Management für einen WWW Cluster

Angenommen eine Gruppe von Web Servern, die nur Anfragen bearbeiten, z.B. Google. Bei großen Providern kann die Web Site aus Hunderten oder Tausenden von Servern bestehen

Vorteil: Einheitliche Anwendung, nur Lesezugriffe zu den HTML Daten.

In diesem einfachen Fall verteilt ein vorgeschalteter Router (auch als Sprayer bezeichnet) die eintreffenden Anfragen nach dem Round Robin Verfahren auf die einzelnen Server.

Beispiel Google: Die Workload Aufteilung ist trivial einfach, weil alle Anfragen identisch sind und nur wenige unterschiedliche Server-seitige Verarbeitungsprogramme benötigt werden. Fast alle Datenbankzugriffe sind Lesezugriffe, und eine **Datenintegrität** der in mehreren duplizierten Datenbanken abgespeicherten Daten ist nicht erforderlich.

# **z/OS Work Load Management Übersicht (1)**

**Die folgenden drei Seiten enthalten eine sehr schöne Übersicht über das z/OS Work Load Management aus Wikipedia ([http://de.wikipedia.org/wiki/Workload\\_Manager](http://de.wikipedia.org/wiki/Workload_Manager)). Wir gehen später auf die hier erwähnten Eigenschaften im Detail ein.**

**Die in jedem Augenblick von einem z/OS System auszuführende Datenverarbeitung wird als „Work“ (Arbeit) bezeichnet. Der einem bestimmten Prozess zuzuordnende Teil davon wird als Work Unit (Arbeitseinheit) bezeichnet. Die in einem bestimmten Augenblick anfallende „Work Load“ ist die Summe aller Work Units.**

**Der Workload Manager (WLM) ist die z/OS Betriebssystemkomponente, die für die Arbeit auf dem Rechner den Zugang zu den Betriebsmitteln steuert. Auf einem Großrechner ist ein prioritisierter Zugang zu Betriebsmitteln (Ressourcen) notwendig, da viele unterschiedliche Anwendungen den Rechner gleichzeitig nutzen und eine den Benutzerwünschen entsprechende Ressourcenverteilung erfolgen muss.**

**Betriebsmittel sind z.B. CPU Zeit, Hauptspeicherplatz, I/O Kanalkapazität oder Plattenspeicherzugriffe. Auch Netzwerk Adapter oder Locks können Ressourcen sein. WLM steuert die Betriebsmittelvergabe auf der Basis von Service Classes (Dienstklassen). Work Units werden den Service Classes über einen Klassifizierungsmechanismus zugeordnet. Die Klassifizierung wird durch den Systemadministrator des z/OS-Systems vorgenommen und kann anhand von Attributen, die für die Programmprodukte unter z/OS existieren, vorgenommen werden. Beispiele für Attribute sind Benutzernamen, Transaktionsnamen, Transaktionsklassen oder Programmnamen, die in den Anwendungen verwendet werden. Als weiteres definiert der Systemadministrator eine Zielvorgabe (Goal) für die Dienstklassen. Die Zielvorgabe kann die durchschnittliche Antwortzeit der Work Units, die in der Klasse laufen, umfassen, einen prozentualen Anteil der Work Units, die in einer bestimmten Zeit enden sollen, oder eine durchsatzorientierte Vorgabe darstellen. Welches Ziel für eine Service Class vergeben werden kann, hängt davon ab, wie viele Informationen der Workload-Manager über die Anwendungen erhält. Neben der Zielvorgabe wird jeder Dienstklasse eine Importance (Wichtigkeit) zugeordnet, die festlegt, welche Klassen bevorzugt bzw. benachteiligt werden sollen, wenn die Betriebsmittel im System nicht mehr ausreichend zur Verfügung stehen.**

## Übersicht (2)

WLM benutzt einen Regelmechanismus, um zur Laufzeit den Zugang zu den Betriebsmitteln zu steuern. Dazu werden kontinuierlich Daten aus dem z/OS-System gesammelt. Dies sind Informationen über die Wartezustände der Work Units auf die Betriebsmittel, die Anzahl der laufenden Work Units und deren Abarbeitungszeiten. Die Informationen werden in Service Classes (Dienstklassen) zusammengefasst entsprechend der Klassifizierung, die durch den Systemadministrator vorgenommen wurde. Dann wird auf der Basis dieser Informationen die Zielerfüllung für jede Klasse berechnet und, falls notwendig, der Zugang zu den Betriebsmitteln angepasst. Die Anpassung erfolgt immer in Abhängigkeit von der Wichtigkeit (importance) der Klassen und dem Grad in dem das Ziel verfehlt wird. Das heißt, die wichtigste Klasse, die am weitesten ihr vorgegebenes Ziel verfehlt hat, wird als erste betrachtet und die Klassen mit der geringsten Wichtigkeit sind die potenziellen Kandidaten, um Betriebsmittel abzugeben. Dabei wird allerdings berücksichtigt, ob ein potenzieller Spender (Donor) auch tatsächlich das benötigte Betriebsmittel verwendet. Dieser Regelmechanismus läuft typischerweise alle 10 Sekunden unter z/OS ab; in der Zwischenzeit werden die Daten für das nächste Berechnungsintervall gesammelt. Ein Berechnungsintervall endet, wenn eine Anpassung zugunsten einer Dienstklasse durchgeführt werden kann.

WLM steuert den Zugang zu den Prozessoren und I/O-Einheiten des Systems, den Zugang zum Speicher und die Bereitstellung von Adressräumen, um Programme für bestimmte Anwendungen abarbeiten zu lassen. Der Zugang zu den Prozessoren wird zum Beispiel über Dispatch Priorities geregelt. Dazu wird allen Arbeitseinheiten einer Dienstklasse (Service Class) dieselbe Priorität zugeordnet, wobei jedoch die Vergabe dieser Priorität nicht in jedem Fall mit der Definition der Wichtigkeit (Importance) der Dienstklasse übereinstimmen muss. Vielmehr orientiert sie sich an der aktuellen Auslastung des Systems, den Anforderungen der Klasse und ihrer Zielerfüllung. Dieses Verhalten des z/OS-WLM nennt man auch zielorientiertes (Goal oriented) Workload-Management, und es ist ein wichtiges Unterscheidungskriterium zu anteilsorientiertem Workload-Management, bei dem feste Zugänge zu den Betriebsmitteln vergeben werden. Letzteres findet sich häufig in Workload-Management-Komponenten von Unix-Systemen, z.B. Sun M9000 Solaris oder Superdome HP-UX.

## Übersicht (3)

**Der WLM setzt umfangreiche adaptive Algorithmen ein um Rechner zu steuern, so dass Zielvorgaben (Business Goals) des Anwenders erfüllt werden. Im Gegensatz zu den Implementierungen auf anderen Plattformen (z.B. HP Superdome, Sun M9000) erfolgt die hier beschriebene Steuerung voll automatisch, ohne Eingriffe durch den Systemadministrator. Es wird ganz auf den Einsatz von vorgefertigten Regeln und von fest einzustellenden Parametern verzichtet. Die Steuerungsalgorithmen passen sich ebenfalls automatisch an das Umfeld an. Die hierfür erforderlichen Anpassungen an eine sich ständig ändernde Workload erfolgen automatisch mit einer Taktrate von z.B. 10 Sekunden.**

**Der zweite essentielle Unterschied des z/OS-WLM zu den Workload-Management Implementierungen auf anderen Plattformen ist die starke Verflechtung mit den Anwendungen und Programmprodukten, die unter einem z/OS-Betriebssystem ablaufen. So ist es durch die ständige Kommunikation zwischen dem WLM und diesen Anwendungen möglich, die Eigenschaften der Anwendungen zu erkennen und im System durch den WLM zu steuern. Dies ist bis dato auf keinem anderen System möglich, in denen jedwede Steuerung auf Prozesse begrenzt ist.**

**Neben der Steuerung eines Systems bietet der z/OS-WLM eine Reihe von Schnittstellen, die es Lastverteilungskomponenten erlauben, Informationen aus dem System zu erhalten, um eine intelligente Verteilung von Arbeit auf eines oder mehrere z/OS-Systeme vorzunehmen. Mehrere z/OS-Systeme können in einem Parallel Sysplex zusammengeschaltet werden, und diese Kombination wird ebenfalls unterstützt, um nach außen ein einheitliches Bild abzugeben. z/OS WLM verfügt außerdem über eine Reihe von weiteren Funktionen, die die Lastverteilung auf einem physischen System zwischen mehreren logischen Systeme unterstützen und den Zugang zu großen Plattenfarmen in Abhängigkeit von der daraufzugreifenden Arbeit steuern.**

**Soweit der Wikipedia Beitrag.**

## Übersicht (4)

### Zusammenfassung:

**Der Work Load Manager (WLM) ist ein z/OS Alleinstellungsmerkmal. Andere Plattformen, z.B. HP Superdome oder Sun M9000, verfügen ebenfalls eine als Work Load Manager bezeichnete Komponente, die vor allem benutzt wird, um virtuelle Maschinen realen CPUs zuzuordnen. Diese haben jedoch bei weitem nicht den Funktionsumfang des z/OS WLM.**

# Begriffe

- Work** andere Bezeichnungen Datenverarbeitung. Data Processing. Ist das, was ein Rechner durchführt.
- Work Unit** ein Stück Datenverarbeitung, z.B. Durchführung einer Transaktion oder eines Stapelverarbeitungsjobs.
- Work Load** =  $\sum$  Work Units , die Summe aller Work Units, die ein Rechner in einem gegebenen Zeitpunkt bewältigt.
- Resourcen** die Betriebsmittel, über die ein Rechner verfügt. Beispiele sind CPU Zyklen, Hauptspeicherplatz. I/O Kanal Kapazität, Plattenspeicherzugriffe
- Service Unit** Ressourcenverbrauch einer Work Unit.
- Service Class** (Dienst Klasse), Gruppe von Work Units mit ähnlichen Vorgaben wie Antwortzeit (Response Time), Turn-around-Time, Priorität, usw.
- Goal** andere Bezeichnung Business Goal, Zielvorgabe für für eine Service Class, z. B. 90 % aller Transaktionen einer bestimmten Service Class sollen eine Response Time < 0,3 Sekunden aufweisen.

Sorry, Sie müssen sich mit den Begriffen vertraut machen.

# Traditionelle Work Load Management Verfahren

Es existieren viele Möglichkeiten und Einstellungsalternativen, die Ausführung der Work Load zu steuern. Beispiele sind:

- Programme laufen Run-to-Completion, oder unterliegen einer Zeitscheibensteuerung
- Die Anzahl der gleichzeitig aktiver Prozesse kann vergrößert oder verkleinert werden, um z.B. jedem Prozess einen optimalen Umfang an realem Speicher zuzuordnen. Prozesse können zeitweise ausgelagert werden, wenn Ressourcen knapp werden.
- Multithreading, Anzahl von Subsystem Instanzen
- In einem Sysplex können Prozesse bestimmten physischen Servern optimal zugeordnet werden
- I/O Kanäle können den LPARs optimal zugeordnet werden
- Prioritäten können angepasst werden
- usw.

# Die Komplexität wächst mit der Systemgröße

Problembereiche sind

- Stapelverarbeitung und interaktive Verarbeitung müssen gleichzeitig laufen
- Belastungsschwankungen treten auf (von Minute zu Minute, während des Tages, innerhalb einer Woche,)
- Die Affinität der Prozesse zu ihren Daten soll optimiert werden
- Die Zuordnung der Prozesse zu realen CPUs muss erfolgen
- Prozesse erzeugen unterschiedliche I/O Anforderungen und I/O Belastungen

In einer IT-Umgebung ohne Mainframe ist es üblich, unterschiedlichen Anwendungen unabhängigen physischen Servern zuzuordnen. Die Folge ist eine schlechte Auslastung der Server (typisch 20 % CPU-Auslastung). Dies erscheint akzeptabel, denn die Hardware ist billiger als bei einer Mainframe Lösung. Jedoch die Folgen sind:

- Viele Server (tausende in großen Unternehmen)
- Heterogene IT-Landschaft
- Komplexe LAN Strukturen
- Hoher Administrationsaufwand
- Hoher Energieverbrauch und Kühlungs-/Klimatisierungsaufwand

Vor allem die hohen Administrationskosten machen diese Lösung zunehmend unattraktiv.

In einer Mainframe Installation sind Energieverbrauch und Kühlungs-/Klimatisierungsaufwand deutlich geringer, da eine CPU Auslastung im 90 – 100% Bereich möglich ist. Dies ermöglicht der z/OS Workload Manager. Weiterhin ist der Administrationsaufwand deutlich geringer, wozu die **Tivoli** und **Unified Resource Manager** Komponenten beitragen.

# System Ressourcen

Unter System Ressourcen versteht man Elemente wie

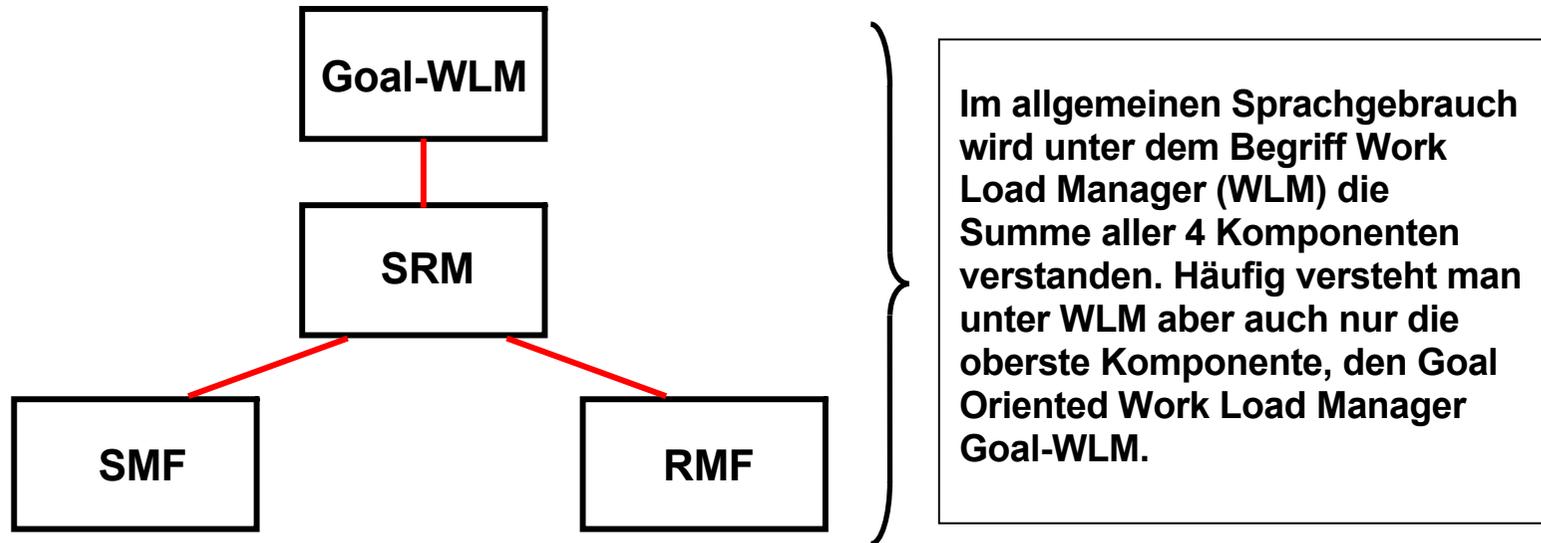
- Verarbeitungskapazität der individuellen CPUs,
- Platz im Hauptspeicher,
- Channel Subsystem Verarbeitungskapazität,
- I/O Kanal Übertragungskapazität,
- Plattenspeicherzugriffe,
- ...

Der Workload Manager hilft, alle System Ressourcen optimal auszunutzen.

**Anmerkung: Die deutsche Sprache verwendet die Schreibweise Ressource. Die englische Schreibweise ist Resource. Wir benutzen den Begriff Ressource/Resource teilweise alleinstehend und teilweise als Teil eines Begriffes, z.B. „System Resource Manager“.**

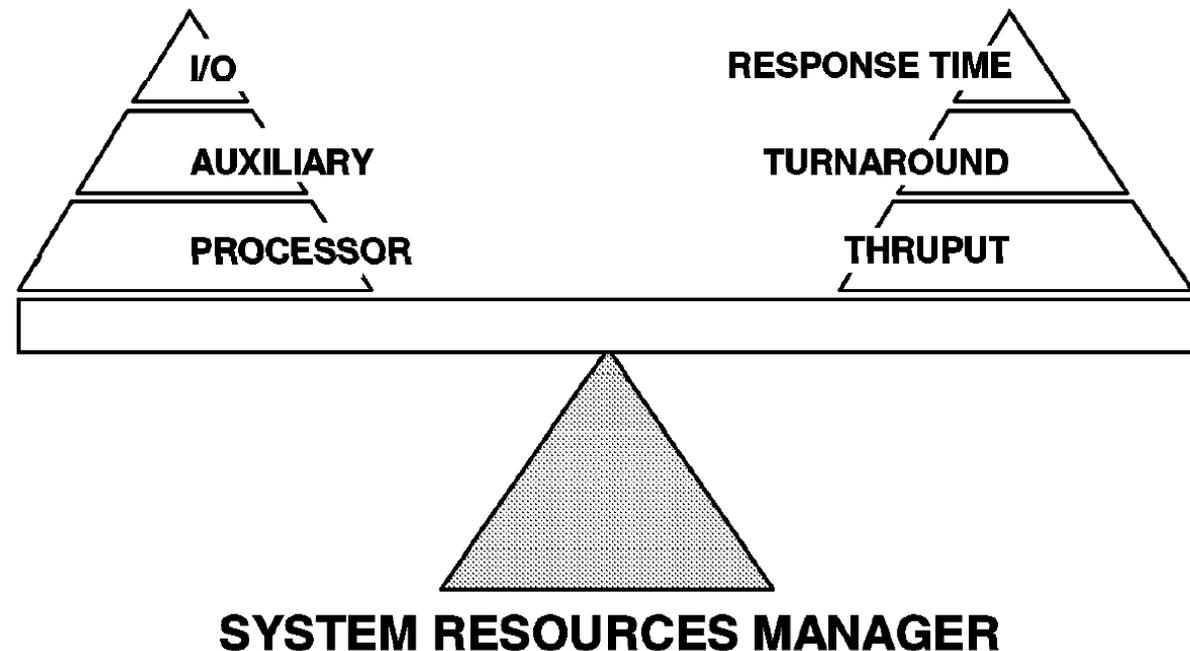
**Die unterschiedliche Schreibweise ist verwirrend. Wir verwenden deshalb ausschließlich die englische Schreibweise.**

**Die Verwendung der deutschen Übersetzung „Betriebsmittel“ ist im Zusammenhang mit WLM eher ungebräuchlich.**



## Es existieren vier z/OS Work Load Manager Komponenten

- Der **Goal oriented Work Load Manager** Goal-WLM stellt die Schnittstelle zum Systemadministrator dar und erlaubt eine voll-automatische Ablaufsteuerung-
- Der **System Resource Manager** (SRM) ist eine Komponente des z/OS Kernels, welche die eigentlichen Anpassungen vornimmt.
- Die **System Management Facility** (SMF) sammelt System-relevante Information über die Konfiguration, die Auslastung der einzelnen Systemkomponenten (z.B. Hauptspeicher, CPUs, paging/swapping Aktivitäten, I/O Aktivitäten usw. Die Ergebnisse werden in SYS1.MANn Data Sets festgehalten.
- Die **Resource Measurement Facility** (RMF) erstellt Durchschnittswerte über festgelegte Zeitintervalle und macht sie als Input für die erwähnte Tivoli Komponente verfügbar.



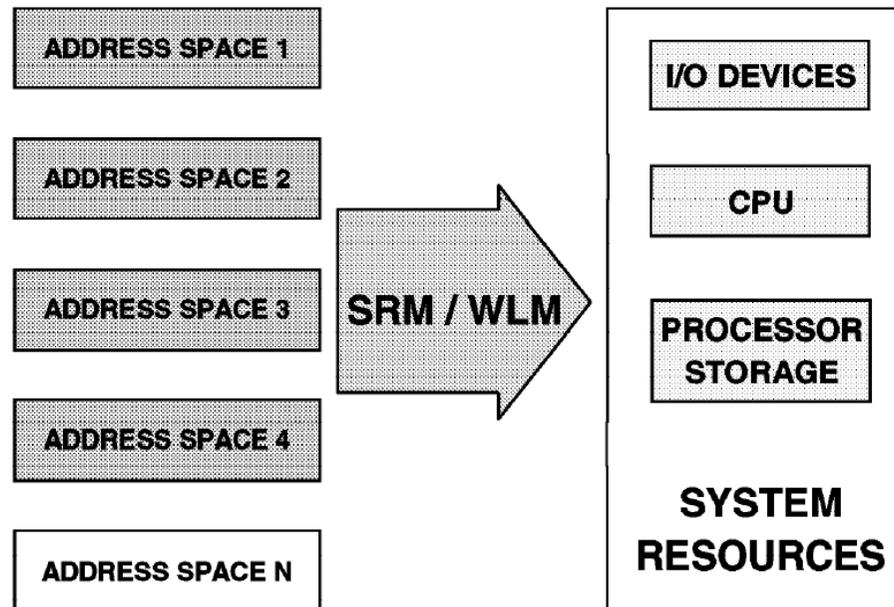
Die System Resource Manager Komponente des Work Load Managers beobachtet für alle angeschlossenen Systeme:

- CPU Auslastung
- Hauptspeicher Nutzung
- I/O Belastung

Der System Resources Manager stellt sicher dass:

- die System Ressourcen möglichst voll genutzt, aber nicht übercommitted werden, und
- alle Benutzer einen fairen Anteil der System Resource erhalten.

SRM überwacht die Nutzung der Ressourcen und ordnet sie periodisch neu zu. Wenn eine Resource schlecht ausgelastet ist, versucht SRM die Systembelastung zu erhöhen. Wenn eine Resource überlastet ist, versucht SRM die Systembelastung zu verringern.



## System Resource Manager

Prozesse sind entweder aktiv (ihnen ist ein Adressenraum mit realem Hauptspeicherplatz zugeordnet), oder sie sind inaktiv (auf einen Plattenspeicher ausgelagert).

SRM bestimmt, welchen aktiven Prozessen (bzw. ihren Adressenräumen) welche System Ressourcen (und wieviel davon) zugeordnet werden. SRM entscheidet dies um zwei grundsätzlich miteinander konkurrierende Ziele zu erreichen:

1. System Resources werden einzelnen Prozessen zugeteilt um die für die Installation festgelegten Ziele in Bezug auf Antwort-Zeit, turnaround-Zeit und Prioritätsanforderungen optimal zu erfüllen
2. System Ressourcen werden optimal genutzt um einen möglichst hohen Systemdurchsatz zu erreichen.

# **System Management Facility SMF**

**Die System Management Facility (SMF) überwacht kontinuierlich, welche Betriebsmittel eine Service Class (Gruppierung von Work Units mit ähnlichen Anforderungen) 1. benutzt und 2. auf welche sie wartet.**

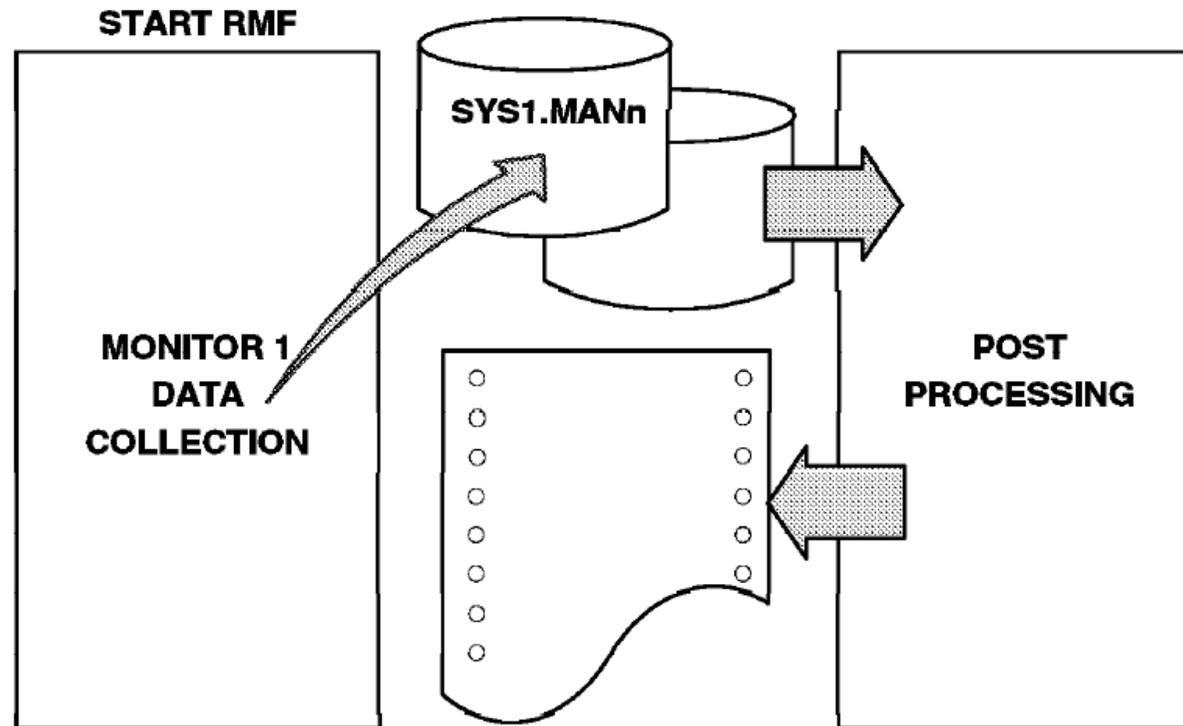
**SMF sammelt Daten über den aktuellen Zustand aller verwalteten Betriebsmittel. Dies sind Information über**

- die Prozessoren,**
- den Speicher und**
- die Nutzung der Platteneinheiten und Kanäle.**

**Aus den gesammelten Daten wird für jede Work Unit festgestellt, ob sie Betriebsmittel benutzt oder darauf wartet. Der Workload-Manager verwendet diese Daten, um den Zugang der Service-Klassen zu den Betriebsmitteln zu regeln.**

**Die System Management Facility (SMF) bewirkt, dass System-bezogene Information über die Konfiguration, die Workload und paging/swapping Aktivitäten in SYS1.MANn Data Sets gesammelt werden.**

**SMF Records des Job Entry Subsystems werden ebenfalls in den SYS1.MANn Data Sets gespeichert. Diese Records enthalten Information über die Start- und Stopzeiten, Prozessor Aktivitäten und Hauptspeichernutzung für jeden Job Step und jede TSO Session. Hilfsprogramme analysieren die Daten und erstellen Berichte.**



## Resource Measurement Facility

Die Resource Measurement Facility (RMF) ist ein eigener Prozess in einem eigenen Adressenraum.

Ein RMF Monitor sammelt System Daten, z.B. CPU Auslastung, DASD Aktivitäten, Belastung der Kanäle, Hauptspeicher Auslastung und Nutzung sowie andere Aktivitäten. RMF erstellt Zusammenfassungen und Berichte hierüber. Der System Administrator bestimmt die Größe der Messintervalle.

# Goal oriented Work Load Manager

**Der System Resource Manager ist auf statische Parameter des System Administrators angewiesen und kann sich nur in geringem Umfang an dynamische Änderungen anpassen.**

**Der Goal Oriented Work Load Manager verfügt über umfangreiche adaptive Algorithmen um die Betriebsmittel abhängig von der sich dynamisch ändernden Systemlast zu steuern, ohne erforderliche Eingriffe durch den Systemadministrator.**

**Der z/OS Goal oriented Work Load Manager (Goal-WLM) erweitert den Funktionsumfang des SRM, und dehnt ihn auf den ganzen Sysplex aus. Er stellt eine Shell für den System Resource Manager (SRM) dar. Goal-WLM ist eine SRM Erweiterung, welche die existierenden SRM Mechanismen für die Zuordnung von Ressourcen zu Work Units um Performance Management Funktionen erweitert.**

**Goal oriented Workload Management stellt eine Veränderung der Fokussierung dar, von**

- **tuning auf der System Resource Ebene, nach**
- **Definitionen des erwarteten Leistungsverhaltens.**

**Diese Eigenschaften sind es, was den z/OS WLM von Work Load Management Funktionen in anderen Betriebssystem unterscheidet.**

# Literatur

Eine sehr gute Übersicht über den Work Load Manager ist zu finden in Kapitel 6 des Lehrbuches

M. Teuffel, R. Vaupel: „Das Betriebssystem z/OS und die zSeries“. Oldenbourg 2004., ISBN 3-486-27528-3.

Die wichtigsten Redbooks sind

z/OS Intelligent Resource Director, August 2001, SG24-5952-00,

<http://www.redbooks.ibm.com/abstracts/sq245952.html>

System Programmer's Guide to: Workload Manager, January 2006, SG24-6472-02

<http://www-ti.informatik.uni-tuebingen.de/~spruth/edumirror/xx044.pdf>

<http://www.redbooks.ibm.com/redbooks/pdfs/sq246472.pdf>

[http://h18004.www1.hp.com/products/quickspecs/11726\\_div/11726\\_div.HTML](http://h18004.www1.hp.com/products/quickspecs/11726_div/11726_div.HTML)