

FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

**Test des CPE-Tools
der Firma SAS**

Monika Trienekens

KFA-ZAM-IB-9416

April 1994
(Stand 09.06.1994)

Inhaltsverzeichnis

Einleitung	v
1 Kernfragen der Untersuchung	1
2 Testergebnisse	3
2.1 Allgemeine Funktionalität	3
2.2 Beispiel-PDB's und Voreinstellungen	4
2.3 Aufbau einer eigenen Testumgebung	5
2.4 Offene Fragen	10
3 Erweiterbarkeit	15
4 Einsetzbarkeit	17
Anhang A Informationsseminar CPE	19
Anhang B Cray-Accounting mit SLR	21

Einleitung

Im Rahmen einer Probeinstallation des Produktes SAS/CPE (SAS / Computer Performance Evaluation) auf IBM RS/6000 unter AIX im Februar 1994 sollte geklärt werden, ob das Produkt geeignet ist, die SLR-Software auf dem IBM-Mainframe unter MVS abzulösen. Hierzu wurde eine Liste mit Fragen zusammengestellt, die im Laufe des Tests geklärt werden und Aufschluß darüber geben sollten, ob SAS/CPE den gestellten Anforderungen gerecht wird.

In dem hier zusammengestellten Testbericht werden zunächst die Kernfragen aufgeführt, die als Leitfaden für den Test dienten. Im nachfolgenden Kapitel wird dann ein Überblick über den Testverlauf mit einer Beantwortung der Fragen gegeben, soweit dies im Rahmen des Tests möglich war. Danach erfolgt ein kurzer Ausblick, inwieweit die Software flexibel genug ist, um den gewünschten Bedürfnissen ausreichend angepaßt zu werden. Abschließend erfolgt ein Resümee, in dem nach Abwägung aller Vor- und Nachteile eine Aussage getroffen wird, in welchem Rahmen SAS/CPE eine Alternative zu dem zur Zeit in Produktion befindlichen SLR darstellen kann.

1 Kernfragen der Untersuchung

Die folgenden Punkte wurden beim Test untersucht:

- Welche Daten sind in den Beispiel-PDBs (performance **d**atabases) enthalten und welche vordefinierten Formen der Auswertung werden mit dem Produkt ausgeliefert? Sind die zugehörigen Tabellen, Reports und Views für uns nutzbar?
- Wie groß ist der Aufwand, eine eigene PDB mit eigenen Tabellen und Reports zu definieren?
- Welche Formen der Präsentation werden geboten (Grafiken, Tabellen, ASCII-Ausgabe zur Weiterverarbeitung)?
- Welche Bedienungsmöglichkeiten gibt es (interaktiv / Batch)? – Inwieweit ist eine Automatisierung möglich?
- Wieviel Speicherplatz / CPU-Zeit wird im allgemeinen benötigt? Welche Systemvoraussetzungen sind notwendig, damit der sonstige Rechnerbetrieb nicht beeinträchtigt wird? Wie stark werden die Daten komprimiert?
- Welche Arten von Reports sind mit den zur Verfügung stehenden Daten möglich (Accounting – Institutsübersichten, Kapazitätsplanung, Ressourcenauslastung)?
- In welchem Format müssen die Eingabedaten vorliegen? Welche Möglichkeiten bestehen, die Daten der verschiedenen Unix-Systeme im Haus (Cray – UNICOS, RS/6000 – AIX, Intel Paragon – OSF/1) sinnvoll zu erfassen? Gibt es bereits Prototypen für Konvertierungsprogramme, die zum Einlesen dieser Daten benötigt werden?
- Wie fehleranfällig ist das System? Ist die Konsistenz der Daten gewährleistet, z.B. nach einem System-/Programmabsturz?
- Welche Backup-Möglichkeiten gibt es (Wiederherstellung alter Zustände, Auslagern alter Daten)?

2 Testergebnisse

In diesem Abschnitt werden die Ergebnisse bzgl. der im vorherigen Kapitel gestellten Fragen, sowie sonstige Merkmale, die beim Testen von SAS/CPE aufgefallen sind, zusammengefaßt.

2.1 Allgemeine Funktionalität

Die SAS/CPE Software für Open Systems stellt in erster Linie ein Hilfsmittel zum Performance-Management und zur Kapazitätsplanung von Computer-Systemen und Netzwerken dar. Es soll System- und Netzwerk-Administratoren dazu dienen, aus der Vielzahl der anfallenden Systemdaten, relativ schnell und einfach Performance- und Kapazitätsanalysen zu erstellen, um so eine bestmögliche Ausnutzung der zur Verfügung stehenden Ressourcen erreichen zu können.

Hierzu bietet SAS/CPE Schnittstellen zu einer zur Zeit noch recht begrenzten Anzahl von Daten-Collectoren. Bei diesen Daten-Collectoren handelt es sich um Software, die auf den zu untersuchenden Rechnern oder Netzwerken Daten sammelt. Diese werden vom SAS/CPE-Tool als Rohdaten eingelesen und interpretiert. Zur Zeit werden von SAS folgende Daten-Collectoren unterstützt:

- SunNet Manager (Sun Microsystems, Inc.)
- Performance Collection Software (Hewlett-Packard Company)
- SPECTRUM (Cabletron Systems, Inc.)
- TRAKKER (Concord Communication, Inc.)
- PROBE/Net (Landmark Systems)
- Standard UNIX Accounting für Solaris, AIX/6000 und HP/UX
(Mit dem Standard-UNIX-Kommando **accton** werden Process Accounting Daten geschrieben, d.h. pro Prozeß Informationen über User, Gruppe, User-Zeit, System-Zeit, Speicherplatzverbrauch, etc., die aber leider keine einheitliche Struktur auf den verschiedenen UNIX-Sytemen haben.)

Zusätzlich zu den direkt im CPE unterstützten Daten-Collectoren bietet die MXG-Software weitere Konvertierungsroutinen (VM, MVS, Cray, ...) zum Umwandeln von Rohdaten in SAS-Format. Sollten auch dort keine geeigneten Konvertierungsroutinen zum Einlesen der gewünschten Daten existieren, besteht die Möglichkeit, beliebige andere Rohdaten einzulesen. Die erforderlichen Definitionen, die SAS/CPE dazu benötigt, die Daten lesen zu können, sind vom Benutzer selbst zu erstellen.

Die Rohdaten werden zunächst in SAS-Format konvertiert und in detaillierter Form in entsprechenden SAS-Dateien abgelegt. Anschließend können diese Daten dann zu Tages-, Wochen-, Monats- und Jahreswerten zusammengefaßt werden, die wiederum in eigenen SAS-Dateien abgespeichert werden. Beim Zusammenfassen der Daten können nicht nur Summen sondern auch Durchschnittswerte, Standardabweichungen oder andere statistische Werte gebildet werden. Auf welcher Ebene (Tag, Woche, Monat, Jahr) welche Daten wie zusammengefaßt und wie lange sie gespeichert werden sollen, wird bei der Definition der Datenbank festgelegt. Durch das Zusammenfassen der Daten können detailliertere Daten, sobald sie nicht mehr benötigt werden, gelöscht werden, wodurch

der Datenumfang stark reduziert werden kann. Das Speichern der Daten erfolgt in sogenannten SAS Daten-Libraries (Subdirectories mit entsprechenden SAS-Dateien), die zusammen *performance database* (PDB) genannt werden. Jede dieser PDBs setzt sich aus neun SAS Daten-Libraries zusammen:

- DETAIL – Daten in detaillierter Form
- DAY – Daten auf Tagesebene
- WEEK – Daten auf Wochenebene
- MONTH – Daten auf Monatsebene
- YEAR – Daten auf Jahresebene
- DICTLIB – Definition und Status der PDB
- drei weitere Libraries zum internen Gebrauch

Die Daten einer PDB unterteilen sich außerdem in verschiedene Tabellen. Jede Tabelle kann dabei eine oder mehrere der oben beschriebenen Datenebenen (Tag, Woche, ...) umfassen. Eine Tabelle ist somit eine logische Gruppierung der Daten über die fünf Daten-Libraries. Tabellen wären beispielsweise Benutzerdaten (verbrauchte CPU-Zeit, Memory, ...) oder Kommandoübersichten (CPU-Zeit, Anzahl Aufrufe, ...). Innerhalb einer PDB können außerdem verschiedene Tabellen über Views miteinander verknüpft werden. Dabei kann wiederum zwischen zwei Arten von Views unterschieden werden:

- *permanent*: Die verknüpften Daten werden in einer neuen Tabelle abgelegt. Die so erzeugten Daten bleiben unverändert, wenn die Daten aus den zugrunde liegenden Tabellen aktualisiert werden.
- *temporär*: Hierbei werden nur die Verknüpfungsarten und nicht die verknüpften Daten gespeichert. Wenn also ein Report auf eine solche View ausgeführt wird, wird jeweils auf die aktuellen Datenbestände in den verschiedenen Tabellen zugegriffen.

Zum Erstellen und Verwalten von PDBs, Tabellen, Reports oder Views besitzt SAS/CPE ein interaktives Interface, über das menügeführt die notwendigen Aktionen ausgeführt werden können. Zusätzlich zu diesem interaktiven Interface gibt es die Möglichkeit, Jobs zum Collectieren von Daten (Laden der Rohdaten und Zusammenfassen der Daten auf die verschiedenen Ebenen), zum Erstellen von Reports oder für anderen Aktionen menügeführt zusammenzustellen und in Dateien zu speichern, damit diese anschließend im Batch-Betrieb ausgeführt werden können.

2.2 Beispiel-PDB's und Voreinstellungen

Zu jedem im SAS/CPE unterstützten Daten-Collector steht eine Beispiel-PDB zur Verfügung, die Daten über einen Zeitraum von einigen Tagen enthält. Außerdem existieren einige vorgefertigte Reports. Die Beispiele, anhand derer die Benutzung von SAS/CPE im Handbuch beschrieben wird, beruhen hauptsächlich auf der Beispiel-PDB, die mit Daten des SunNet Managers gefüllt ist. Entsprechend enthält diese PDB verschiedene Tabellen auf Basis der SunNet Manager Daten und es existieren mehrere passende Reports.

Da während des Tests keines der speziellen Softwarepakete zum Collectieren der Daten zur Verfügung stand, beschränkten sich die Testmöglichkeiten auf das Interpretieren der Daten, die mit dem Standard-UNIX Account-Kommando **accton** erzeugt werden. Zu

diesem Daten-Collector existiert leider nur eine Beispiel-PDB, die HP/UX Daten enthält, zu der es allerdings keine vordefinierten Reports gibt.

Aus diesem Grund konnten die zur Verfügung stehenden Beispiele für den eigentlichen Test nicht weiter genutzt werden, sondern waren nur beim Klären prinzipieller Bedienungsfragen hilfreich.

Beim Anlegen einer neuen PDB sind aber zu jedem unterstützten Machinentyp (also auch für AIX/RS6000) Defaulteinstellungen vorhanden, die bei der Auswahl der benötigten Daten helfen sollen. Diese Voreinstellungen eignen sich als Einstieg, um einen Überblick zu bekommen, welche Möglichkeiten das System bietet. Bei einem Einsatz des Produktes im Produktionsbetrieb ist allerdings sehr genau zu überlegen, welche Werte wirklich benötigt werden. Für diesen Zweck sind die Voreinstellungen wenig hilfreich, da sie nur eine ganz allgemeine Auswahl der Daten darstellen.

2.3 Aufbau einer eigenen Testumgebung

Um SAS/CPE mit eigenen Daten testen zu können, wurde zunächst einmal eine eigene PDB definiert. Diese PDB bestand aus einer Tabelle in der Standard-Accounting Daten einer RS/6000 auf Tages-, Wochen-, Monats- und Jahresebene gesammelt werden konnten.

Das Anlegen der neuen PDB war mit Hilfe des Handbuches ohne größeren Aufwand möglich, da alle Definitionen, die nötig waren, menügeführt abgesetzt werden konnten. Bei der Auswahl der Daten, die in die PDB aufgenommen werden sollten (z.B. CPU-Zeit, Memory, etc.), konnten die Voreinstellungen, die vom System gesetzt waren, weitgehend übernommen werden, da sie für Testzwecke ausreichend waren. Gleiches galt für die Festlegung der Ebenen, auf denen sie gespeichert werden sollten (täglich, wöchentlich, monatlich, jährlich) sowie für die Auswahl der Statistiken, die für die einzelnen Werte erzeugt werden sollten (Summe, Durchschnitt, etc.).

Diese PDB konnte nun mit Daten gefüllt werden. Als Testdaten wurden hierzu die Accounting-Daten einer RS/6000 Mod.220 über einen Zeitraum von zehn Tagen gesammelt. Bei dieser Maschine handelt es sich um einen Client innerhalb der ZAM-internen AIX-Gruppe, so daß der Umfang der anfallenden Daten relativ gering war. Zum Collectieren dieser Daten wurde ein Batchjob erstellt. Die einzelnen Bestandteile dieses Jobs (Laden der Rohdaten und Reduzieren der Daten auf die verschiedenen Ebenen) konnten wiederum über eine interaktive Menüführung zusammengestellt werden. Die zugehörigen Macroaufrufe wurden dabei vom CPE erstellt und konnten in Files abgespeichert werden. In einem abschließenden Schritt mußten diese Teilkomponenten dann nur noch mit Hilfe eines Standard-Editors zusammengefügt und der Aufruf des %CPSTART-Macros ergänzt werden, der unter anderem die Angabe der PDB enthält, in die die Daten eingefügt werden sollen. Mit diesem Job konnten die Daten dann im Batch-Mode collectiert werden. Die Arbeitsschritte zum Anlegen einer PDB und Collectieren der Daten waren aufgrund der Menüführung und der vorgegebenen Beispiele im Handbuch ohne größeren Aufwand und detaillierte Kenntnisse des Systems (SAS/CPE) durchführbar.

Anhand dieser PDB konnte nun untersucht werden, welche Darstellungsmöglichkeiten SAS/CPE bietet. Die möglichen Darstellungsformen reichen von den verschiedensten Grafiken (X-Y-Graph, Balken-, Tortendiagramme, etc.) über verschiedene Tabellenformen bis hin zur Kombination mehrerer Grafiken oder Tabellen auf einer Seite. Die

Ausgabe kann sowohl für Grafiken als auch für Tabellen wahlweise in einem ASCII- oder Grafik-Fenster ausgegeben oder in entsprechenden Files gespeichert werden. Das Zusammenstellen der Reports erfolgt wieder über das interaktive Interface. Der zugehörige Programmcode sowie der erzeugte Output können in SAS-Katalogen abgespeichert und jederzeit wieder geladen werden. Soll ein Report nicht über das interaktive Interface gestartet werden, gibt es außerdem die Möglichkeit, den zugehörigen Programmcode in einem externen File abzulegen, so daß der Report im Batch-Betrieb gestartet werden kann. Sowohl für das Erstellen einer PDB oder eines Reports als auch für das Collectieren von Daten bietet SAS/CPE eine gelungene Kombination aus interaktiver Nutzung und Batchbetrieb. Über das interaktive Interface können Reports zusammengestellt und direkt getestet werden. Gleichzeitig ist hierüber auch die Möglichkeit gegeben, Batchjobs zusammenzustellen, ohne die benötigten SAS-Macros zu kennen. Auf der anderen Seite können die interaktiv entwickelten Reports und Collect-Jobs, die zum Teil recht zeitintensiv sein können, aber auch im Batchmode ausgeführt werden, so daß eine Automatisierung regelmäßig anfallender Aufgaben, wie z. B. das tägliche Collectieren der Daten ohne weiteres möglich ist.

Zusätzlich zu der Möglichkeit, Reports oder andere Jobs interaktiv zusammenzustellen, kann ein in SAS erfahrener Benutzer die Jobs jederzeit erweitern oder umstellen, aber auch die Daten mit anderen SAS-Tools weiterverarbeiten, da die Daten in SAS-Libraries abgelegt sind, und somit von jedem anderen SAS-Tool gelesen werden können.

Bei den ersten Tests, die lokal auf einer RS/6000 Mod.220 der ZAM-internen AIX-Gruppe durchgeführt wurden, stellte sich schnell heraus, daß die Maschine nicht dazu ausgelegt war, effektiv mit SAS/CPE arbeiten zu können. Es kam zu folgenden Problemen:

- Beim Erstellen einer Tabelle über das interaktive Interface gibt es die Möglichkeit, im entsprechenden Panel aus sechs vorgegebenen Tabellentypen einen auszuwählen. Lief SAS/CPE lokal auf der RS/6000 Mod.220, erschien beim Versuch, den Tabellentyp zu ändern, die Fehlermeldung *Page Space Low captured in system CPE* und SAS/CPE war anschließend in einem undefinierten Zustand, in dem keine Befehle mehr eingegeben werden konnten.
- Die Antwortzeiten, die lokal auf einer RS/6000 Mod. 220 (Client in der AIX-Gruppe) erzielt wurden, waren verständlicherweise schlechter als die im zentralen AIX (RS/6000 Mod.550), wie die folgende Gegenüberstellung zeigt. Es wurde jeweils derselbe Tagessatz einer RS/6000 Mod.220 collectiert. Dabei ergaben sich im Durchschnitt folgende Zeitmessungen:

Maschine, auf der SAS/CPE gestartet wird	Real-Zeit (SAS-Start)	CPU-Zeit (SAS-Start)	Real-Zeit (Collect-Step)	CPU-Zeit (Collect-Step)
zentrales AIX	7 sec	1:42sec	8:44 min	2:54 min
RS/6000 Mod.220	17:21 sec	3 sec	9:24 min	5:18 min

Da die Voraussetzungen im zentralen AIX verständlicherweise besser waren, wurden die weiteren Tests dort ausgeführt, während die Daten auf dem Server der ZAM-internen AIX-Gruppe gehalten wurden, da dort ausreichend Plattenplatz zur Verfügung stand.

Da es zunächst nicht möglich war, die Tagesdaten des zentralen AIX zu komprimieren – was nach einem Fehler im CPE oder **accton** Kommando aussah – wurden die Messungen, die nötig waren, um Aussagen über Performance und Speicherbelegung zu machen, mit Hilfe der Daten, die auf einem Client der ZAM-internen AIX-Gruppe (RS/6000 Mod.220) gesammelt wurden, durchgeführt. Um eine gewisse Vorstellung zu bekommen, welche Ressourcen zum Verarbeiten der Accounting-Daten verbraucht wurden, werden hier einige Messungen zu CPU-Zeit und Speicherplatzbelegung aufgeführt.

Die nachfolgende Tabelle zeigt exemplarisch die Speicherplatzbelegung der anfallenden Rohdaten, die in einem Zeitraum von 0:00 Uhr bis 14:30 Uhr eines beliebigen Tages auf den verschiedenen Systemen mit dem **accton** Kommando erzeugt wurden:

RS/6000 Mod.220 (Client in der AIX-Gruppe)	zentrales AIX (ZAM112)	Cray Y-MP M94
22.6 KB	6.8 MB	26.2 MB

Im Vergleich dazu hat ein File, das die Rohdaten der Cray Y-MP M94 enthält, die zur Zeit vom SLR eingelesen werden, ein Datenvolumen von ca. 1.04 MB. Das Reduzieren des Volumens der Rohdaten erfolgt auf der Cray über das Cray System Accounting. Dabei handelt es sich um ein den jeweiligen Bedürfnissen anzupassendes Programmpaket, das aus den Standard-Accountingdaten und einigen weiteren Informationen die benötigten Werte herauszieht und zusammenfaßt. Das Accounting für die Cray sollte nach Möglichkeit auch weiterhin auf den Daten des Cray System Accounting und nicht auf den Standard-Accountingdaten aufsetzen, nicht nur weil das Datenvolumen wesentlich geringer ist, sondern auch weil alle benötigten Informationen in diesen Daten enthalten sind, während dies bei den Standard-Accountingdaten auf der Cray nicht der Fall ist.

Die nächste Tabelle gibt eine Übersicht über die CPU-Zeit, die benötigt wird, um die Rohdaten eines Tages einer RS/6000 Mod.220 (Client in der AIX-Gruppe) zu collectieren sowie über die Höhe des Speicherplatzverbrauchs der SAS-Libraries, in denen die Daten gespeichert werden.

Datenvolumen der zu collectierenden Daten eines Tages	Real-Zeit des Collect-Jobs (Einlesen und Reduzieren der Daten)	CPU-Zeit des Collect-Jobs (Einlesen und Reduzieren der Daten)	Speicherplatz- belegung der SAS-Libraries
38.5 KB	7:14 min	2:40 min	1.8 MB (Daten von 10 Tagen)

Da die hier aufgeführten Messungen sich nur auf die Verarbeitung einer sehr kleinen Datenmenge beziehen, sind sie leider nicht sehr aussagekräftig. Dennoch ist aber abzusehen, daß sowohl der Speicherplatzbedarf als auch der CPU-Zeitverbrauch zum Verarbeiten der Daten der zentralen Rechnersysteme im ZAM erheblich sein wird. Daher dürfte es auf alle Fälle sinnvoll sein, die anfallenden Datenmengen nicht unnötig

aufzublähen. Das Cray System Accounting bietet hierzu einen sinnvollen Ansatz, indem es aus den verschiedenen Rohdaten die benötigten Werte in einem File (Superrecord) zusammenfaßt. Auf anderen UNIX-Systemen ist diese Möglichkeit leider nicht gegeben. Deshalb muß bei den dort anfallenden Daten genauestens überlegt werden, welche Daten wirklich benötigt werden, um so die anfallenden Daten zu reduzieren. Zusätzlich sollte die Zeitspanne, über die die Rohdaten im DETAIL-Level gespeichert werden, so gering wie möglich gehalten werden (z.B. ein oder zwei Tage bis sie zu Tages-, Wochen-, Monats- und Jahressummen zusammengefaßt sind). Auch hierdurch kann der Datenumfang erheblich reduziert werden.

Bei dem oben beschriebenen Problem, daß sich die Standard Accounting Daten des zentralen AIX zwar in detaillierte SAS-Daten umwandeln lassen, dann aber nicht weiter komprimiert werden können, während dies bei den Daten, die lokal auf einem Client in der ZAM-internen AIX-Gruppe erzeugt wurden, ohne Probleme möglich war, handelt es sich weder um einen Fehler im CPE noch um einen Fehler im **accton** Kommando. Der Grund für das unterschiedliche Verhalten liegt darin begründet, daß im zentralen AIX erheblich mehr Daten anfallen. Aus diesem Grund wird dort stündlich überprüft, ob der **pacct**-File, in dem die Accounting-Daten gesammelt werden, eine bestimmte Größe überschritten hat. Ist dies der Fall, so wird der aktuelle **pacct**-File abgeschlossen und ein neuer eröffnet, so daß pro Tag mehrere dieser Files erzeugt werden. Auf dem Client in der ZAM-internen AIX-Gruppe dagegen fallen so wenig Daten an, daß nur ein File pro Tag geschrieben wird.

SAS/CPE speichert nun beim Collectieren die Daten zunächst im DETAIL-Level und erst wenn dort Daten über einen längeren Zeitraum (mehrere Stunden) vorliegen, werden sie für die höheren Level zusammengefaßt und dort eingefügt. Da im zentralen AIX die einzelnen **pacct**-Files meist einen kleineren Zeitraum umfassen, müssen zunächst mehrere dieser Files collectiert werden, bevor die Daten in die Tages-, Wochen-, Monats- oder Jahresebene eingetragen werden. Nachdem dieses Problem zufriedenstellend gelöst war, war es aber dennoch nicht möglich, die Daten des zentralen AIX zu reduzieren, da der entsprechende Batchjob auf einen I/O-Error lief und abbrach. Dieses Problem ist ausführlich in der Liste der noch offenen Fragen beschrieben.

Da es sich bei den Daten, die mit dem **accton** Kommando erzeugt werden, um Process Accounting Daten handelt, könnten anhand dieser Daten für die bisher im ZAM noch nicht kontingentierte Rechnersysteme wie das AIX eine Accounting-Formel bestimmt werden, über die die Nutzung abgerechnet wird. Das Cray-Accounting sollte aber nach Möglichkeit – wie oben beschrieben – besser auf den Daten des Cray System Accountings aufgesetzt werden, da im Standard UNIX Accounting auf der Cray nicht alle Informationen, die zur Abrechnung benötigt werden, vorhanden sind.

Zum Umsetzen der oben erwähnten Accounting-Formeln bietet SAS/CPE die Möglichkeit, zusätzlich zu Variablen, deren Werte eingelesen werden, weitere Variablen zu definieren, deren Werte aus den eingelesenen berechnet werden. Dadurch müssen keine Änderungen in den Rohdaten vorgenommen werden, sondern es können zusätzliche Variablen (z.B. für Verrechnungs- oder Kontingenteinheiten) definiert werden, deren Werte aus den zur Verfügung stehenden Rohdaten berechnet werden.

Wie im vorigen Abschnitt beschrieben, handelt es sich bei den mit **accton** erzeugten Daten um Process Accounting Daten, die für Aussagen über Kapazitätsplanung und Ressourcenauslastung wenig Informationen liefern. Für diese Bereiche liefern die im ersten Kapitel aufgeführten Daten-Collectoren die entsprechenden Daten. Allerdings konnte bis jetzt noch nicht geklärt werden, welche der Daten-Collectoren auf welchen UNIX-Systemen einsetzbar sind.

Will man also mit SAS/CPE Daten zur System- oder Netzwerkauslastung und Kontrolle verwalten, so sollten Überlegungen angestellt werden, ob einer der von SAS unterstützten Collectoren einsetzbar wäre. Eventuell ist es aber auch günstiger, sich eine Schnittstelle zum SAS/CPE selbst zu definieren, über die Daten aus diesen Bereichen eingelesen werden können, die über entsprechende Standard UNIX-Kommandos erzeugt werden, die zur Systemüberwachung zur Verfügung stehen.

Wie sich in den ganzen vorherigen Beispielen zeigt, bietet SAS/CPE für einen Einsatz auf den zentralen Rechnersystemen des ZAM zur Zeit nur sehr eingeschränkte Möglichkeiten, Daten zu collectieren, da auf diesen Rechnern keiner der unterstützten Daten-Collectoren installiert ist und auch die MXG-Software zur Zeit nicht verfügbar ist. Außerdem beziehen sich die von den unterstützten Collectoren gesammelten Daten überwiegend auf Kapazitäts- und Ressourcenauslastung und nur ganz begrenzt auf das User-Accounting. Aus diesem Grund wäre eine Grundvoraussetzung, um SAS/CPE sinnvoll für das Accounting der zentralen Rechner im ZAM einsetzen zu können, daß entweder die MXG-Software die benötigten Konvertierungsroutinen bietet oder aber die Möglichkeit besteht, auch andere als die von den unterstützten Daten-Collectoren erzeugten oder mit MXG konvertierbaren Daten in SAS/CPE einzulesen. Laut SAS ist diese Möglichkeit gegeben. Zur Zeit muß der Benutzer die hierzu benötigten SAS-Macros in der SAS-Programmiersprache „per Hand“ erstellen. Diese Macros definieren die Struktur der Rohdaten, so daß diese in der gewünschten Form eingelesen und in SAS-Daten umwandelt werden können. Anschließend können die Daten mit SAS/CPE wie die von den Collectoren erzeugten Daten auf die verschiedenen Ebenen komprimiert werden. Für den nächsten Maintenance-Level von SAS/CPE (Frühjahr '94) ist angekündigt, daß eine zusätzliche Komponente integriert sein soll, die es ermöglicht, entsprechende Definitionen über das interaktive Interface vorzunehmen.

Leider konnte das Umwandeln von Rohdaten in SAS-Daten über eigene Macros und die anschließende Weiterverarbeitung dieser Daten aus Zeitgründen und fehlenden Kenntnissen der SAS-Programmiersprache nicht weiter getestet werden, so daß wir uns hier zunächst nur auf die Aussagen der Firma SAS stützen können.

Während des gesamten Tests, der in den letzten Wochen durchgeführt wurde, hat sich gezeigt, daß das System SAS/CPE relativ stabil ist. Es kam zwar aufgrund von Bedienungsfehlern oder fehlenden Ressourcen einige Male dazu, daß das System in einem undefinierten Zustand abgebrochen werden mußte, dabei blieben in der Regel aber immer alle Dateninformationen erhalten. Verweise waren dafür aber im CPE teilweise nicht mehr bekannt. So existierte in einem Fall z.B. keine aktive PDB mehr. In diesem Fall konnte der Ausgangszustand relativ schnell wiederhergestellt werden, indem die Referenzen neu definiert wurden. Anschließend konnte wieder ohne Probleme auf die alten Daten zugegriffen werden. Probleme gab es allerdings als ein Collect-Job beim

Reduzieren der Daten im DETAIL-Level auf Tages-, Wochen-, Monats- und Jahresebene auf einen I/O-Error lief. Der Job wurde daraufhin automatisch abgebrochen, so daß die PDBs anschließend in einem inkonsistenten Zustand waren, da auf den höheren Ebenen noch nicht alle Werte eingefügt worden waren. Dieser Zustand konnte im CPE nicht rückgängig gemacht werden. Es wurde beim anschließenden Erstellen eines Reports auch keine Meldung ausgegeben, daß die Daten ggf. inkonsistent sein könnten. Die einzige Möglichkeit, in den Ausgangszustand zurückzukommen, scheint darin zu bestehen, sich jeweils ein Backup des alten Zustandes zu halten, um dieses im Fehlerfall wieder einspielen zu können.

SAS/CPE bietet leider keine eigenen Backup-Möglichkeiten. Da es sich bei SAS-Libraries aber um einfache Directories handelt, besteht die Möglichkeit, mit der im ZAM zur Verfügung stehenden Software selber Backups zum Sichern der Datenbestände zu erstellen.

2.4 Offene Fragen

Leider blieben nach Abschluß des Tests noch einige Fragen und Probleme offen, die noch nicht vollständig geklärt werden konnten:

- Das Collectieren der Daten des zentralen AIX auf DETAIL-Ebene (Umwandeln der Rohdaten in SAS-Daten) läuft ohne Probleme. Sobald diese Daten allerdings im REDUCE-Step reduziert werden sollen, tritt ein I/O-Error auf und der Job bricht ab (reproduzierbar).

Im Log-File wird folgende Fehlermeldung ausgegeben:

```
ERROR: An I/O error has occurred on file PDBWORK.REDUCE1.DATA.  
ERROR: File PDBWORK.REDUCE1.DATA is damaged. I/O processing  
did not complete.
```

Hierbei fiel außerdem auf, daß die temporären Files im Directory *pdbwork* sehr viel Speicherplatz benötigen. So belegten die Work-Files zum Zeitpunkt des Jobabbruchs ca. 21 MB, während die zu reduzierenden Daten im DETAIL-Level nur ein Datenvolumen von 10 MB hatten.

Lösung: Der Grund für diesen Fehler ist vermutlich ein zu kleiner Swap-Space. Er sollte bei etwa 150MB mindestens aber 100MB liegen. Der SAS/CPE-Test wurde aber unter einer „normalen“ Userid ohne besondere Privilegien durchgeführt, der nur ein begrenzter Swap-Space, der unterhalb des erforderlichen Minimums liegt, zugeteilt wird. Leider erfolgte die Lösung dieses Problems durch die Firma SAS erst Anfang Juni, so daß ein direkter Test, ob der Fehler durch ein Einräumen entsprechender Privilegien für die Test-Userid wirklich behoben ist, leider nicht mehr möglich war, da die Testlizenz bereits abgelaufen war.

- Welche Hardware-Voraussetzungen sind erforderlich, um auf einer RS/6000 effektiv mit SAS/CPE arbeiten zu können?

Außerdem ergaben sich während des Tests noch einige Fragen bzw. Anmerkungen, die aus Benutzersicht von Bedeutung für die Nutzung von SAS/CPE sind. Diese Punkte wurden auch bereits mit Mitarbeitern der Firma SAS besprochen. Die nachfolgende Liste umfaßt die Fragen mit den von SAS erhaltenen Antworten und den sich daraus ergebenden Konsequenzen:

- **Problem:** Im interaktiven Interface scheint es keine Möglichkeit zu geben, beim Erstellen eines normalen X-Y-Plots von Seiten des Benutzers Einfluß auf die Gestaltung der Achsen zu nehmen (z.B. Positionierung der Achsenbeschriftung oder Wahl eigener Label).

Bei Ausgabe der Grafik in einem ASCII- statt Grafik-Window wurde sogar die vom Benutzer gesetzte Achsenbeschriftung ignoriert und eine defaultmäßig gesetzte ausgegeben.

Antwort: Es ist im SAS/CPE nur möglich, die vertikale Achse eines X-Y-Plots zu modifizieren. Es soll aber noch geklärt werden, ob dieser Zustand endgültig ist, oder ob Änderungen an dieser Stelle zu erwarten sind.

Konsequenz: Sollte es dabei bleiben, daß nur die vertikale Achse verändert werden kann, besteht hierin eine sehr starke Einschränkung der grafischen Darstellungsmöglichkeiten im CPE selbst. Bei einer Jahresübersicht würde es beispielsweise keine direkte Möglichkeit geben, an der X-Achse die Monatsnamen statt der zugehörigen Zahlenwerte auszugeben. Die einzige Möglichkeit besteht dann darin, die Änderungen nachträglich per Hand in den mit SAS/CPE erstellten Source einzufügen, falls eine Änderung der entsprechenden Parameter dort möglich ist und sie nicht fest gesetzt sind. Ist eine Änderung der Parameter nicht möglich, bleibt nur die Möglichkeit, die Grafik direkt mit einem anderen SAS-Tool zu erzeugen.

- **Problem:** Wenn eine Tabelle, die über mehrere Seiten geht, im Grafik-Format ausgegeben wird, wird jeweils nur die erste Seite der Tabelle ausgegeben. Auf alle weiteren Seiten ist kein Zugriff möglich. Die Grafik enthält nur eine Fußnote, daß die Tabelle nicht vollständig ist. Im ASCII-Mode dagegen werden alle Seiten der Tabelle ausgegeben.

Antwort: Diese unvollständige Darstellung einer Tabelle wurde von SAS als Fehler anerkannt und wird entsprechend weitergeleitet.

- **Problem:** Wenn ein Report auf Monatsebene erstellt wird, ist dennoch die Möglichkeit gegeben, im entsprechenden Panel für den gewünschten Zeitraum, über den der Report erstellt werden soll, einen Wert anzugeben, der nicht auf Monatsgrenzen liegt (z.B. 4.2.94 – 10.2.94).

Wird ein solcher „falscher“ Zeitraum angegeben, scheinen die Ergebnisse wirklich die Werte über diesen angegebenen Zeitraum darzustellen. Der Definition nach, sollten aber eigentlich auf Monatsebene nur Monatssummen verfügbar sein. Von daher erscheint es wenig sinnvoll, auch andere Zeitangaben vom System her zu erlauben, da dies ggf. leicht zu Bedienungsfehlern führen kann.

Antwort: Auch auf Monatsebene hat die DATETIME-Variable noch alle Werte, d.h. sie ist nicht auf Monatsebene reduziert, sondern enthält noch alle Angaben vom Jahr, Monat, Tag bis hin zur Stunden- und Minutenangabe. Ob dies so gewünscht ist, war beim Technical Support in Heidelberg auch nicht näher bekannt, es soll aber noch geklärt werden.

Konsequenz: Wenn es wirklich so vorgesehen ist, daß man bei der Angabe der gewünschten Zeitspanne auch kleinere Zeiträume als die auf dem angesprochenen Level eigentlich verfügbaren angeben kann, liegt hierin für den Benutzer eine mögliche Fehlerquelle, daß z.B. durch Tippfehler falsche Grenzen eingegeben werden (z.B. für den gesamten Januar 1.1. – 30.1.). Um solchen Fehlern vorzubeugen, wäre es wesentlich sinnvoller, wenn beispielsweise auf Monatsebene nur Jahr und Monat und nicht auch noch Tag, Stunde und Minute angegeben werden dürften. Auch würde dies der Intention der Reduzierung der Daten auf verschiedene Level entsprechen, daß nämlich auf Monatsebene auch nur Monatswerte verfügbar sind.

- **Problem:** In Grafiken können die gewünschten Daten über die verschiedensten Zeitspannen dargestellt werden, z.B. über mehrere Tage. Dabei werden in der Grafik die Tagessummen der einzelnen Werte dargestellt. Werden dieselben Werte allerdings in Tabellenform dargestellt, werden keine Tagessummen, sondern mehrere Einzelwerte pro Tag (zu jedem einzelnen Wert der für die DATETIME Variable existiert) ausgegeben. Dabei entspricht der ausgegebene Zeitraum noch nicht einmal dem vom Benutzer vorgegebenen Zeitintervall, sondern erstreckt sich scheinbar über den gesamten Zeitraum, für den die entsprechenden SAS-Tabellen Daten enthalten.

Antwort: Da die DATETIME Variable, wie im vorherigen Punkt beschrieben, nicht entsprechend reduziert wird, sondern weiterhin alle Zeitangaben enthält, ist es korrekt, daß in diesem Fall auch alle Einzelwerte statt des Summenwertes über den gesamten Tag ausgegeben werden. Daß die ausgegebenen Werte nicht mit dem angegebenen Zeitraum übereinstimmen, konnte nicht nachvollzogen werden.

Konsequenz: Sollte es keine Möglichkeit geben, die Werte der DATETIME-Variablen bei der Ausgabe in Tabellenform entsprechend aufzulösen, d.h. daraus beispielsweise den Monat oder das Jahr herauszuziehen, um so einen Gesamtwert einer Variablen über den entsprechenden Zeitraum ausgeben zu können, bedeutet dies eine erhebliche Einschränkung der Nutzbarkeit des Tools, denn genau das sollte doch eigentlich der Sinn der verschiedenen Level sein, daß auf Monatsebene nur auf Monatssummen und nicht auf kleinere Zeiträume zurückgegriffen werden kann, denn hierfür existieren schließlich die entsprechenden unteren Level (Tag oder Woche).

- **Problem:** Ein Report, der auf Monatsebene in einem X-Y-Plot den gesamten CPU-Verbrauch pro Monat darstellen soll, sollte als Ergebnis einen Plot liefern, der pro Monat jeweils einen Wert enthält. Liegen diesem Report jetzt z.B. nur Daten zugrunde, die innerhalb eines Monats gesammelt wurden (z.B. Februar, dieser aber auch noch nicht komplett), sollte die Grafik auch nur aus einem einzigen Wert bestehen. Statt dessen enthält der Plot zwei Werte für die X-Koordinaten 1 und 5 (Januar/Mai oder 1./5.Februar ?).

Antwort: Für den Fall, daß die Tabellen auf Monatsebene z.B. nur Daten vom 1. und 5. Februar enthalten, wird in den zugehörigen SAS-Dateien aus der Datumsvariablen nur der Wert des Tages herausgezogen und auf eine temporäre Datei geschrieben. Entsprechend enthält diese anschließend zwei Werte, nämlich 1 und 5.

Konsequenz: In dem beim Testen aufgetretenen Fehlerfall sind zuvor die Daten von mehr als zwei einzelnen Tagen collectiert worden. Zudem erscheint es wenig sinnvoll – wie auch bereits in den vorherigen Punkten angesprochen – wenn auf Monatsebene Tagessummen gebildet werden. Stattdessen sollte das Ergebnis eine Zwischensumme

des bis dahin berechneten Monatswertes darstellen, da ansonsten frühestens nach Abschluß eines Monats die Möglichkeit besteht, auf die entsprechenden Monatswerte zuzugreifen. In der Praxis besteht aber der Bedarf, bereits während des laufenden Monats, aktuelle Aussagen über die bis dahin gesammelten Werte zu machen.

Es wäre wünschenswert, auf diese hier aufgeführten Fragen definitive Antworten zu bekommen, d.h. ob die hier angesprochenen Punkte auf Fehlern in SAS/CPE oder auf Bedienungsfehlern beruhen, oder ob bestimmte Interpretationen genau so wie beschrieben auch vorgesehen sind. Denn nur wenn diese Punkte eindeutig geklärt sind, kann eine definitive Aussage getroffen werden, ob SAS/CPE sinnvoll eingesetzt werden kann.

3 Erweiterbarkeit

Mit Hilfe des interaktiven Interface werden beispielsweise Jobs zum Collectieren von Daten oder zum Erstellen von Reports erstellt. Anschließend kann der zugehörige SAS-Source sozusagen als Prototyp in einem File abgespeichert und bei Bedarf per Hand nacheditiert werden. Außerdem steht der Source von allen CPE-Macros zur Verfügung, wodurch die Möglichkeit besteht, in Anlehnung an diese, eigene Macros zu erstellen, die den jeweiligen Anforderungen entsprechend angepaßt sind. Dies wäre eventuell bei der Darstellung der Daten in Tabellenform nützlich, da CPE nur sechs fest definierte Tabellentypen anbietet, die nicht unbedingt immer den Anforderungen genügen müssen. Außerdem können mit Hilfe der SAS-Programmiersprache eigene Schnittstellen zum Einlesen von Rohdaten zu erstellt werden, so daß der Benutzer hier nicht nur auf die von SAS vorgegebenen Schnittstellen angewiesen ist, sondern prinzipiell beliebige Daten verarbeiten kann.

Insgesamt gibt es an vielen Stellen die Möglichkeit, die Nutzung des SAS/CPE-Tools zu erweitern bzw. auf andere SAS-Tools zurückzugreifen. Allerdings sind hierzu ausreichende allgemeine SAS-Kenntnisse (z.B. Basic Language, Funktionalität anderer SAS-Tools, ...) erforderlich.

4 Einsetzbarkeit

Die allgemeine Funktionalität ist auf den ersten Blick recht zufriedenstellend. Aufgrund der Menüführung ist SAS/CPE an den meisten Stellen – wenn die entsprechenden allgemeinen Vorkenntnisse vorhanden sind – leicht verständlich. Mit allgemeinen Datenbankkenntnissen und dem zur Verfügung stehenden Handbuch sind die Datenbank-spezifischen Aufgaben (Anlegen einer PDB, Daten collectieren) ohne großen Aufwand zu bewältigen. Das Erstellen von Reports und Grafiken bedarf allerdings einiger Gewöhnung, da die Menüführung in diesem Bereich nicht unbedingt intuitiv ist. Außerdem sind die Darstellungsmöglichkeiten in einigen Bereichen – wie die oben aufgeführten Probleme zeigen – recht eingeschränkt, was für den Benutzer beim Erstellen einer Grafik aber nicht unbedingt direkt ersichtlich ist.

Wie man hieran sieht, ist die Verarbeitung der Daten im SAS/CPE zu Statistiken, Grafiken etc. zwar recht vielfältig aber noch nicht unbedingt zufriedenstellend, wie die während des Tests aufgetretenen Probleme zeigen. Da sich die Darstellungsmöglichkeiten der Daten nicht nur im CPE, sondern auch über andere SAS-Tools realisieren lassen, sollten in diesem Bereich aber ausreichende Möglichkeiten für die verschiedenen Darstellungsformen gegeben sein.

Da SAS/CPE zur Zeit nur eine sehr begrenzte Anzahl an Daten-Collectoren unterstützt, bleibt zu untersuchen, inwieweit die MXG-Software zusätzliche Konvertierungsroutinen bietet, die genutzt werden können (Cray), bzw. wie aufwendig es ist, Daten über selbstgeschriebene Schnittstellen in SAS-Format umzuwandeln, um sie dann im CPE weiterverarbeiten zu können (Paragon). Denn nur in diesem Fall wäre das Produkt sinnvoll einsetzbar.

Außerdem bleibt zu untersuchen, wie groß der Datenumfang auf den zentralen Rechnersystemen des ZAM ist, und ob für diese Datenmengen ein Einsatz von SAS/CPE effektiv genug ist.

Insgesamt hat sich gezeigt, daß SAS/CPE in Kombination mit anderen SAS-Tools prinzipiell die Grundfunktionalität bietet, die benötigt wird, um das Accounting damit ausführen zu können. Allerdings müßte noch erheblich Arbeit investiert werden, um die Software endgültig einsetzen zu können, da die Vorgaben von SAS nur ansatzweise genutzt werden könnten, ein Teil der Schnittstellen zum Einlesen von Daten aber selbst erstellt oder modifiziert werden müßte. Und auch das Layout der Reports müßte an mehreren Stellen verändert und angepaßt werden.

Außerdem ist die Art der Datenkomprimierung auf den verschiedenen Leveln – wie die offenen Fragen zeigen – nicht unbedingt einsichtig.

Zusammenfassend kann gesagt werden, daß SAS/CPE durchaus eine Alternative zum SLR darstellen kann.

Jedoch kann nach dem vierwöchigen Test noch keine abschließende Entscheidung getroffen werden, da aufgrund der aufgetretenen Fehler wichtige Fragen nicht beantwortet werden konnten. Mit einer verbesserten robusten Version und unter Einbeziehung der MXG-Konvertierungsroutinen muß ein weiterer Test durchgeführt werden.

Anhang A Informationsseminar CPE

Am 8. März 1994 fand in Köln ein Informationsseminar der Firma SAS statt, auf dem das neue Tool CPE vorgestellt wurde.

Dabei ergaben sich zur Funktionalität des Produktes selber wenig Neuigkeiten, dafür aber vielleicht einige allgemeine Informationen, die von Interesse sein könnten:

- In Deutschland gibt es das Produkt SAS/CPE seit Oktober '93 als Testversion und seit Januar '94 als Produktionsversion.
- Zur Zeit gibt es in Deutschland drei Testinstallationen der SAS/CPE-Software, wovon die KFA die erste war, die mit den Tests begonnen hat. Wo die beiden anderen Testinstallationen vorgenommen wurden, ist leider nicht bekannt.
- SAS ist bemüht, die Liste der Daten-Collectoren, die von der CPE-Software unterstützt werden, weiter auszubauen. Für das zweite Quartal 94 ist NetView 6000 (von IBM) angekündigt und es werden Überlegungen angestellt, demnächst einen Daten-Collector für Cray-Daten zu unterstützen. Dabei ist an eine Integration der MXG-Software gedacht, die scheinbar schon eine Schnittstelle von Cray-Daten zur CPE-Software auf anderen Systemen (z.B. MVS) bietet.
Zu diesem Zweck wäre SAS an einem Gespräch mit uns interessiert, um zu erfahren, welche Ansprüche wir in diesem Fall an das System stellen würden und inwieweit diese realisierbar sind.
- SAS legt Wert darauf, daß sie dem Benutzer keine fertige Produktionsversion vorlegen, sondern ihm das nötige Know-how vermitteln, damit er die Möglichkeit hat, die Software jederzeit seinen individuellen Bedürfnissen anzupassen. Sie leisten aber auch vor Ort Hilfestellung, um zusammen mit dem Benutzer eine für ihn geeignete Lösung zu finden.

Anhang B Cray-Accounting mit SLR

Bei der Ablösung des SLR durch eine UNIX-Alternative bleibt von den zur Zeit ausgewerteten Systemen (VM, MVS, Cray) nur das Cray-Accounting, das auf das neue System portiert werden muß, während die anderen Systeme durch die Ablösung des IBM-Mainframe automatisch wegfallen.

Um eine gewisse Vorstellung zu haben, welche Cray-Daten mit dem SLR verwaltet werden, die auch mit einem Nachfolgeprodukt (möglichst ohne allzugroße Umstellungen) auswertbar sein sollten, und um einen Anhaltspunkt zu haben, welche Daten auch für eine Auswertung anderer UNIX-Systeme von Interesse sein könnten, sollen im folgenden die Daten aufgeführt werden, die zur Zeit auf beiden Crays gesammelt und ausgewertet werden.

- **User- / System-Accounting und Job-Klassen**

Die Daten werden auf der Cray mit dem an unsere Bedürfnisse angepaßten Cray System Accounting erzeugt (csafef). Die dort erzeugten Daten werden folgendermaßen direkt gespeichert:

Name	Erläuterung	log	user	system	class
eingelezene Werte					
RECTYP	job record type (I–interactive, B–batch, R–rexec, F–ftp, U–unknown)	X			
PRIOR	user (selected) priority (0, 1, 2, 4, 6)	X	key	key	key
USERID	userid	key	key	key	last
JOBID	jobid	X			
GROUP	group of CRAY userid	X	last		last
CLASS	job class / queue name (e.g. M04P1)	X			key
REQNAME	request name (e.g. of a batch job)	X			
SYSID	system identification (CRAY / HLRZ)	key	key	key	key
STARTTIME	job start time	X			
ENDTIME	job termination time	X			
TURNTIME	endtime – starttime (sec)	X			
SEQNO	job sequence number	key			
QUEUEWAIT	user queue wait time (sec)	X	sum	sum	sum
CPUUSR	CPU user time (sec)	X	sum	sum	sum

Name	Erläuterung	log	user	system	class
CPUSYS	CPU system time (sec)	X	sum	sum	sum
CPU	CPU time (sec) (system + user)	X	sum	sum	sum
IOWAIT	IO-wait time (sec)	X	sum	sum	sum
MEMINT	memory integral (kwords*sec)	X	sum	sum	sum
AVGMEM	average memory (kwords)	X	max	max	
MEMFACT	memory factor	X			
NOBLOCKS	I/O blocks	X			
NOPHYSIO	# of physical I/O requests	X	sum	sum	
NOLOGIO	# of logical I/O requests	X	sum	sum	
FEBLOCKS	# of blocks transferred to front end	X	sum	sum	
MULTI	multitasking breakdown (*100)	X			
VE	system billing units	X	sum	sum	sum
KE	contingent units	X	sum	sum	sum

feste Parameter					
INST	account group	X	key	key	last
ACCOUNT	user account	X	key		
COSTID	costid (I–intern, E–extern, H–HLRZ, U–Uni, S–System)	X	last	last	last
UNAME	user name	X	last		
PERIOD	shift and/or weekend	X			

berechnete Werte					
TAT	total turnaround time: turntime + queuwait	X	sum	sum	sum
TYP_I	interactive: count(rectyp=I)	X	sum	sum	sum
TYP_B	batch: count(rectyp=B)	X	sum	sum	sum
TYP_R	rexec: count(rectyp=R)	X	sum	sum	sum

Name	Erläuterung	log	user	system	class
TYP_F	ftp: count(rectyp=F)	X	sum	sum	sum
TYP-U	unknown: count(rectyp=U)	X	sum	sum	sum
KES_0	if prio=0 : ke	X	sum	sum	
KES_L	if prio=1 : ke	X	sum	sum	
KES_M	if prio=2 : ke	X	sum	sum	
KES_H	if prio=4 : ke	X	sum	sum	
KES_I	if prio=6 : ke	X	sum	sum	
VES_0	if prio=0 : ve	X	sum	sum	
VES_L	if prio=1 : ve	X	sum	sum	
VES_M	if prio=2 : ve	X	sum	sum	
VES_H	if prio=4 : ve	X	sum	sum	
VES_I	if prio=6 : ve	X	sum	sum	
JOB_P0	if prio=0 : 1 else : 0	X	sum	sum	
JOB_PL	if prio=1 : 1 else : 0	X	sum	sum	
JOB_PM	if prio=2 : 1 else : 0	X	sum	sum	
JOB_PH	if prio=4 : 1 else : 0	X	sum	sum	
JOB_PI	if prio=6 : 1 else : 0	X	sum	sum	
NOJOBS	count(endtime)		X	X	X
PRICE_FAC	if costid=[I,U,H] : 1 if costid=E : 2 if costid=S : 0		X		
DM_P0	if costid=i : ve*price_fac else : ke*price_fac		X		
DM_PL	if costid=i : ve*price_fac else : ke*price_fac		X		
DM_PM	if costid=i : ve*price_fac else : ke*price_fac		X		
DM_PH	if costid=i : ve*price_fac else : ke*price_fac		X		

Name	Erläuterung	log	user	system	class
DM_PI	if costid=i : ve*price_fac else : ke*price_fac		X		
DM_TOT	if costid=i : ve*price_fac else : ke*price_fac		X		
CPU_MIN	cpu/60				X
CPU_HRS	cpu/3600		X	X	X
TAT_MIN	tat/60				X
TAT_HRS	tat/3600		X	X	X
AVGCPU	cpu/nojobs		X	X	X
AVGCPUUSR	cpuusr/nojobs		X	X	X
AVGCPUSYS	cpusys/nojobs		X	X	X
AVGMEM	memint/cpu		X	X	X
AVGIOWAIT	iowait/nojobs		X	X	X
AVGNOPHYSIO	nophysio/nojobs		X	X	
AVGNOLOGIO	nologio/nojobs		X	X	
AVGFEBLOCKS	feblocks/nojobs		X	X	
AVGQUEUEWAIT	queuwait/nojobs		X	X	X
AVGTAT	tat/nojobs		X		X
ETMF	tat/cpu		X	X	X
ICPU	if sysid=CRAY : 4 if sysid=HLRZ : 8		X		
MFCPU	cpu/icpu		X		
MFCPUHRS	cpu_hrs/icpu		X		

- **Kommando-Statistik**

Die Daten für die Kommandostatistiken werden ebenfalls über eine im Haus erstellte Erweiterung im csafef erzeugt. Dabei werden die folgenden Daten im SLR gespeichert:

Name	Eräuterung	log	cmd
eingelezene Werte			
SYSID	system identification (CRAY / HLRZ)	key	key
RECIND	record identification 'CMD'	—	—
RECTYP	all commands 'ALL' or only interactive 'INTER'	key	key

Name	Eräuterung	log	cmd
TIME	date and time (MMDDYYhhmmss)	key	
CMD	command name	key	key
NUMBER	# of command counts	X	sum
CPU	CPU time (sec)	X	sum
ELAP	elapse time (sec)	X	sum
ELAP_MIN	minimum elapse time	X	min
ELAP_MAX	maximum elapse time	X	max
MEMORY_INT	memory integral	X	sum
BLOCKS	# of blocks transferred	X	sum
LOG_IO	logical I/O	X	sum
PHYS_IO	physical I/O	X	sum

berechnete Werte			
AVG_CPU	cpu/number	X	X
AVG_ELAP	elap/number	X	X
AVG_MEM	memory_int/cpu	X	X
AVG_BLOCKS	blocks/number	X	X
AVG_LOGIO	log_io/number	X	X
AVG_PHYSIO	phys_io/number	X	X
MAXMEM	max(avg_mem)		X
CPU_HRS	cpu/3600		X
ELAP_HRS	elap/3600		X
ELAP_MAX_HRS	elap_max/3600		X

- **Hardware Performance Monitor**

Der Hardware Performance Monitor unter UNICOS (HPM) liefert Informationen, wieviel Operationen ein Job in welcher Zeit gerechnet hat. Aus diesen Daten stellt ein im Haus erstelltes FORTRAN-Programm die zur MFlop-Berechnung relevanten Informationen zusammen und legt sie in SLR verständlicher Form im MVS ab. Dabei werden die folgenden Werte im SLR gespeichert:

Name	Erläuterung	log	ops
eingeleseene Werte			
SYSID	system identification (HLRZ / CRAY)	key	key
RECTYPE	record identification 'MVOPS'	—	—

Name	Erläuterung	log	ops
HPMTIME	job HPM time (MMDDYYhhmmss)	key	
USERID	userid	key	key
GROUP	group of userid	X	X
CPU	CPU time (sec)	X	sum
VINTOPS	number of vector integer & logical operations	X	sum
VFLOATOPS	number of vector floating operations	X	sum

feste Parameter			
INST	account group	X	key
COSTID	costid (I–intern, E–extern, H–HLRZ, U–Uni, S–System)	X	X

berechnete Werte			
VOPS	vintops+vfloatops	X	sum
NOJOBS	count(hpmtime)		X
CPU_HRS	cpu/3600		X‘
MIOPS	(vintops/1000000)/cpu		X
MFLOPS	(vfloatops/1000000)/cpu		X
MVOPS	(vops/1000000)/cpu		X
AVGCPU	cpu/nojobs		X

- Verfügbarkeit**

Die auf der Cray über system activity (sa) erzeugten Daten zur Verfügbarkeit des Rechners werden im VM so aufbereitet und in IBM-internes Format umgewandelt, daß sie vom SLR eingelesen werden können:

Name	Erläuterung	log	avl
eingelesene Werte			
SYSID	system identification 'HLRZ' / 'CRAY'	key	key
REC_ID	record identification 'AVL'	—	—
TIMESTAMP	date and time (MMDDYYhhmmss)	key	
DAY_TIME	seconds per day	X	sum
PROD_TIME	production time	X	sum
NO_IPL	number of production ipls	X	sum
BREAK	number of breaks (not used)	—	—

Name	Erläuterung	log	avl
NO_HARD_BRK	number of hardware breaks	X	sum
NO_SOFT_BRK	number of software breaks	X	sum
NO_UNKN_BRK	number of unknown breaks	X	sum
T_HARD_BRK	hardware break time (sec)	X	sum
T_SOFT_BRK	software break time (sec)	X	sum
T_UNKN_BRK	unknown break time (sec)	X	sum
T_HARD_SERV	hardware service time (sec)	X	sum
T_SOFT_SERV	software service time (sec)	X	sum
T_UNKN_SERV	unknown service time (sec)	X	sum
berechnete Werte			
NO_DAYS	count(timestamp)		X
PROD_HRS	prod_time/3600		X
H_BRK_HRS	t_hard_brk/3600		X
S_BRK_HRS	t_soft_brk/3600		X
U_BRK_HRS	t_unkn_brk/3600		X
BREAK_HRS	h_brk_hrs+s_brk_hrs+u_brk_hrs		X
H_SRV_HRS	t_hard_srv/3600		X
S_SRV_HRS	t_soft_srv/3600		X
U_SRV_HRS	t_unkn_srv/3600		X
SERV_HRS	h_srv_hrs+s_srv_hrs+u_srv_hrs		X
DOWN_HRS	serv_hrs+break_hrs		X
AVG_IPL	ipl/no_days		X
BREAK	h_brk+s_brk+u_brk		X
AVG_BREAK	break/no_days		X
UPTIME (in %)	(prod_time/day_time)*100		X
MTBF	mean time between failure: if (break \geq 1) : (prod_hrs+break_hrs)/(break+1) else : no value		X