REGIONALES HOCHSCHULRECHENZENTRUM
KAISERSLAUTERN

RHRK

# Auftragsverbund mit automatischer Lastverteilung im Kaiserslauterer Lastverbundsystem

Martin Bürkle
Dieter Lunk

RHRK-Nr. 7803

PFAFFENBERGSTRASSE 95, D-6750 KAISERSLAUTERN, TEL.: (0631) 205-2263

# Inhaltsverzeichnis

**Einleitung** ..... **3**

**1. Funktionsbeschreibung** ..... **4**

**2. Aufgaben des KL&KOOP am Eingaberechner** ..... **5**

2.1 Auswahl des optimalen Arbeitsrechners (automatische Lastverteilung) ..... 5

2.2 Transfer der Auftragsinformation zum ausgewählten Arbeitsrechner ..... 8

**3. Aufgaben des KL&KOOP am Arbeitsrechner** ..... **9**

3.1 Kreation des Sekundärauftrags ..... 9

3.2 Auftragsbegleitinformation ..... 9

3.3 Transfer der Ergebnisinformation ..... 9

**4. Aufgaben des KL&KOOP am Ausgaberechner** ..... **9**

**5. Allgemeine Aufgaben des KL&KOOP** ..... **10**

Dieter Lunk
Dr. Martin Bürkle

1978

# Einleitung

Innerhalb des am Regionalen Hochschulrechenzentrum Kaiserslautern (RHRK) entwickelten Verbundsystems nimmt der Auftragsverbund eine zentrale Stellung ein. Er beinhaltet zum einen den Transfer von Aufträgen zu beliebigen Rechnern innerhalb des Netzes; die diesbezügliche Auswahl erfolgt mit Hilfe der automatischen Lastverteilung. Dabei wird aus einer vom Benutzer vorgegebenen Menge von potentiellen Arbeitsrechnern derjenige ausgewählt, der bezüglich der im Lastverteilalgorithmus definierten Kriterien als optimal erscheint.

Zum anderen ist dafür Sorge zu tragen, dass am Arbeitsrechner anfallende Ergebnisinformation von transferierten Aufträgen (Ablaufprotokoll, Druck-, Stanzaufträge, u.a.) an den Eingaberechner oder, falls vom Benutzer explizit gewünscht, an einen beliebigen Netzrechner transferiert wird.

Die im Rahmen des Auftragsverbunds anfallenden Aufgaben obliegen dem Modul KL&KOOP, der als Permanenzauftrag in jedem Netzrechner genau einmal vorhanden ist. Zur Wahrnehmung dieser Aufgaben kommuniziert er über Warteschlangen mit Benutzeraufträgen, mit anderen Verbundkomponenten am eigenen Rechner und -mit Hilfe des Transportsysteme- mit den KL&KOOP's in den angeschlossenen Partnerrechnern.

Die Programmiersprache von KL&KOOP ist COBOL. Die in COBOL für die Manipulation der Auftragsinformation und die Kommunikation zwischen Aufträgen fehlenden bzw. nur unzureichend vorhandenen Sprachmittel sind in Form von Assembler- bzw. BCPL-Unterprogrammen realisiert. Innerhalb des am Regionalen Hochschulrechenzentrum Kaiserslautern (RHRK) entwickelten Verbundsystems nimmt der Auftragsverbund eine zentrale Stellung ein. Er beinhaltet zum einen den Transfer von Aufträgen zu beliebigen Rechnern innerhalb des Netzes; die diesbezügliche Auswahl erfolgt mit Hilfe der automatischen Lastverteilung. Dabei wird aus einer vom Benutzer vorgegebenen Menge von potentiellen Arbeitsrechnern derjenige ausgewählt, der bezüglich der im Lastverteilalgorithmus definierten Kriterien als optimal erscheint.

Zum anderen ist dafür Sorge zu tragen, dass am Arbeitsrechner anfallende Ergebnisinformation von transferierten Aufträgen (Ablaufprotokoll, Druck-, Stanzaufträge, u.a.) an den Eingaberechner oder, falls vom Benutzer explizit gewünscht, an einen beliebigen Netzrechner transferiert wird.

Die im Rahmen des Auftragsverbunds anfallenden Aufgaben obliegen dem Modul KL&KOOP, der als Permanenzauftrag in jedem Netzrechner genau einmal vorhanden ist. Zur Wahrnehmung dieser Aufgaben kommuniziert er über Warteschlangen mit Benutzeraufträgen, mit anderen Verbundkomponenten am eigenen Rechner und -mit Hilfe des Transportsysteme- mit den KL&KOOP's in den angeschlossenen Partnerrechnern.

Die Programmiersprache von KL&KOOP ist COBOL. Die in COBOL für die Manipulation der Auftragsinformation und die Kommunikation zwischen Aufträgen fehlenden bzw. nur unzureichend vorhandenen Sprachmittel sind in Form von Assembler- bzw. BCPL-Unterprogrammen realisiert.

# 1. Funktionsbeschreibung

Als erstes durchläuft KL&KOOP die Initia1isierungsphase, in der im wesentlichen der Aufbau interner Listen sowie ein Informationsaustausch mit allen anderen Partner-KL&KOOP's im Netz erfolgt. Da konzeptionell nur gleichberechtigte Netzrechner vorhanden sind (kein Master-Slave-Prinzip!), wird dadurch insbesondere gewährleistet, dass an allen Rechnern etwa über Gerätekonfiguration (Geräteverbund), Netztopologie, gegenwärtige bzw. wahrscheinliche künftige Belegungssituation (automatische Lastverteilung) usw. jeweils aller anderen angeschlossenen Rechner Kenntnis besteht.

Danach tritt KL&KO0P in seinen zentralen Wartezustand ein, aus dem er ggf. von seinen diversen Auftraggebern (Benutzeraufträge, andere Verbundkomponenten) geweckt wird. Er durchläuft sodann den der jeweiligen Aufgabenstellung zugeordneten Programm-Modul und kehrt wieder in den zentralen Wartezustand zurück.

Die verschiedenartigen Aufgaben des KL&KO0P werden im folgenden beschrieben, wobei bezüglich des Auftragsverbunds zwischen den Funktionen am Eingabe-, Arbeits- bzw. Ausgaberechner unterschieden wird.

# 2. Aufgaben des KL&KOOP am Eingaberechner

## 2.1 Auswahl des optimalen Arbeitsrechners (automatische Lastverteilung)

Es wird ausgegangen von einem Rechnernetz

$$N = \{C_1, .., C_n\}$$

bestehend aus Rechnern $C_i$.

Weiterhin wird für jeden $C_i \in N$ eine Teilmenge $N_i \subseteq N$

$$N_i := \{C_{i_1}, \dots, C_{i_{m_i}} \mid 1 \leq i \leq n\}$$

definiert, die aus denjenigen Rechnern besteht, die mit $C_i$ gekoppelt sind.

Die grundsätzliche Forderung besteht nun darin, für jeden an irgendeinem $C_i \in N$ ankommenden, vom Benutzer als Verbundauftrag gekennzeichneten Auftrag A, einen "optimalen" Arbeitsrechner $C_j \in N_i$ auszuwählen ($C_j$ nicht notwendig verschieden von $C_i$), den Auftrag dorthin zu transferieren und die Ergebnisformation an $C_i$ oder an einem vom Benutzer angegebenen Rechner $C_k \in N_j$ auszugeben.

Die Kennzeichnung als Verbundauftrag erfolgt durch die Angabe des Kommandos

@VERBUND, RNR=<Liste von Rechner-Nr.>@.

als erstem Kommando.

Damit definiert der Benutzer eine für den Auftrag A zulässige Menge $Z_A \subseteq N$, deren Rechner für die Bearbeitung von A überhaupt in Frage kommen. In weiteren Spezifikationen können zusätzliche Angaben gemacht werden, die aber für den grundsätzlichen Ablauf der Lastverteilung nur von untergeordneter Bedeutung sind.

Unter diesen Voraussetzungen als "optimal" für den Auftrag A wird nun derjenige Rechner $C_j$ betrachtet, für den gilt:

a) $C_j \in T_A = Z_A \cap Ni$
b) die Betriebsmittelanforderungen von A verletzen keine der momentan gültigen Betriebsmittelgrenzen an $C_j$, wodurch sich $T_A$ zu $T_{A'} \subseteq T_A$ reduziert
c) $C_j$ läßt unter Berücksichtigung der Wichtigkeit von A bezüglich der auf $C_j$ vorhandenen und künftig noch zu erwartenden Aufträge die geringste Bearbeitungsdauer für A erwarten.
d) A kann in "zumutbarer" Zeit an $C_j$ bearbeitet werden, falls $C_j = C_i$.

Da die Entscheidung über den Transfer von A am Eingaberechner $C_i$ getroffen wird, muss die dafür benötigte Information, die die Belegungs- und Betriebssituation aller $C_j \in N_i$ beschreibt, an $C_i$ vorhanden sein. Dazu ist notwendig, dass an jedem am Netz beteiligten Rechner $C_K \in N$ diese sogenannte Statusinformation in mindestens periodischen Zeitabständen gesammelt und an alle Partnerrechner $\in N_K$ versandt wird. Es ist evident, dass die daraus gewonnenen Aussagen umso genauer sind, je aktueller die zur Verfügung stehende Statusinfo ist.

Da für einen an $C_i$ eintreffenden Auftrag A, dessen Bearbeitungszeit an $C_j$ berechnet werden soll, eine "Beschreibung" der Abarbeitungstrategie an $C_j$ also Teil der Statusinfo sein mussmuss, sei folgender Formalismus eingeführt:

Für jeden Rechner $C_j \in N$ sei eine gewisse, möglicherweise von j abhängige, Anzahl $R_j$ von Prioritätsklassen definiert. Ein an $C_j$ eintreffender Auftrag A wird nun einer dieser Prioritätsklassen zugeordnet, etwa der K-ten, wofür dann geschrieben wird:

$$P_j(A) ::= K, K \in \{1,2,...R_j\}$$

Weiterhin sei vereinbart: je kleiner K, desto höher ist die Priorität von A. Ein Auftrag A mit $P_j(A) = K$ wird erst bearbeitet, wenn kein Auftrag B mit $P_J(B) < K$ vorhanden ist.

Aufträge mit gleicher Priorität werden nach dem first-come-first-served-Prinzip abgearbeitet.

Die Beschreibung der Abarbeitungsstrategie eines Rechners reduziert sich damit im wesentlichen auf die Zuordnung eines Auftrags zu einer bestimmten Prioritätsklasse, also auf relativ wenige "griffige" Werte.

Es soll nun dargestellt werden, wie unter Berücksichtigung der oben definierten Optimalitätskriterien der Arbeitsrechner ausgewählt wird, d.h., da die Punkte a) und b) nur relativ triviale Untersuchungen bedingen, wie sich die Bearbeitungsdauer am Rechner $C_j \in T_{A'}$ eines zum Zeitpunkt T am Rechner $C_i$ eintreffenden Auftrags A mit einem voraussichtlichen Rechenzeitbedarf $\rho_A$ errechnet.

Es sei $\varepsilon_k^{(j)}(t) \geq 0$ die Belastungsintensität durch Aufträge A mit $P_j(A) = k$ am Rechner $C_j$, also ein Maß für den Auftragseingang in einer Prioritätsklesse pro Zeiteinheit.

Weiter sei $0 \leq M^{(j)}(t) \leq 1$ die Entlastungsintensität, ein Maß für die CPU-Aufnahme durch Aufträge.

Es erscheint sinnvoll, beide Größen als zeitabhängig anzusehen, da sie so als tageszeitbedingt sehr unterschiedliche Verhaltensweisen der Benutzer und der Betriebsorganisation am besten widerspiegeln.

Wenn A nach $C_j$ transferiert werden würde, dann wäre

$$t_\alpha^{(j)}(A) = t + \tau_{IJ}^{(E)}(A)$$

der Ankunftszeitpunkt bzw. der Bearbeitungsbeginn auf $C_j$, wenn $\tau_{IJ}^{(E)}(A)$ die Transferzeit von $C_i$ nach $C_j$ beschreibt.

Bezeichnet

$$\rho_{k,s}^{(j)} = \sum_{p_j(B) \leq k} L_B$$

man die zum Zeitpunkt $t_s$ an $C_j$ gemessene Summe der vorhandenen Lasten aller Aufträge B mit $P_j(B) \leq k$, so ist, wenn noch

$$\delta_k^{(j)}(t) = \sum_{\mu=1}^{k-1} \varepsilon_\mu^{(j)}(t)$$

gesetzt wird,

$$\rho_{k,\alpha}^{(j)} = \rho_{k,s}^{(j)} + \rho_A + \int_{t_s}^{t_{\alpha^{(j)}}(A)} \left(\delta_{k+1}^{(j)}(t) - M^{(j)}(t)\right) dt \qquad (1)$$

die voraussichtliche Belastung aus Aufträgen B mit $P_j(B) \leq k$ zur Zeit $t_\alpha^{(j)}(A)$ an $C_j$.

Entsprechend der vereinbarten Bearbeitungsstrategie ergibt sich der Bearbeitungsendzeitpunkt $t_\omega^{(j)}(A)$ von A auf $C_j$ nun gerade daraus, dass $p_{k,\alpha}^{(j)}$ und alle bis zum Zeitpunkt $t_\omega^{(j)}(A)$ eintreffenden Aufträge B mit $P_j(B) < k$ abgearbeitet werden, so dass sich $t_\omega^{(j}(A)$ aus

$$\rho_{k,\alpha}^{(j)} + \int_{t_\alpha^{(j)}(A)}^{t_\omega^{(j)}(A)} \left(\delta_k^{(j)}(t) - M^{(j)}(t)\right) dt = 0 \qquad (2)$$

errechnet.

Bezeichnet $\tau jk(A)$ die Transferzeit der Ausgabeinformation A von Cj nach Ck, so ist der optimale Rechner nun entweder der Eingaberechner Ci, falls

$$C_i \in T'_A \text{ und } t_\omega^{(i)}(A) + \tau_{jk}(A) \leq S(\varrho_A, T)$$

wobei $S(\varrho_A,T)$ die in d) eingeführte, von $\varrho_A$ abhängige, zumutbare Schranke bedeutet, oder der Rechner $C_j \in T'_A$, für den

$$t_\omega^{(i)}(A) + \tau_{jk}(A)$$

minimal wird.

**Statusinformation**

Die an $C_j \in N$ gesammelte und an alle $C_i \in N_j$ verschickte Statusinfo hat folgenden Aufbau:

a) Zeitpunkt der Messung (ts)
b) Zuordnungsvorschrift von Aufträgen zu Prioritätsklassen
c) Die augenblickliche Belastung ($\varrho_{k,s}^{(j)}$, $k=1,\ldots,R_j$)
d) Momentan gültige maximale Betriebsmittelgrenzen
e) $N_j$ zum Aufbau einer Netzmatrix in jedem Rechner
f) Be- und Entlastungsintensitäten werden als stückweise konstante Funktionen betrachtet, so dass sich die Integrale in (1) und (2) zur Summation von Rechteckflächen reduzieren.

Die $\varepsilon_k^{(j)}(t)$ und $M^{(j)}(t)$, die sich aus Messungen ergeben haben, sind fest, sollen aber in einer weiteren Ausbaustufe nur noch als Kerninformation fungieren, die KL&KOOP entsprechend den realen Gegebenheiten jeweils mit Anpassungskorrekturen versieht.

## 2.2 Transfer der Auftragsinformation zum ausgewählten Arbeitsrechner

Nach Durchlaufen des Lastverteilalgorithmus teilt KL&KOOP dem Benutzerauftrag den optimalen Rechner mit. Ist dies nicht der eigene Rechner, so beginnt jetzt der Transfer der Auftragsinformation zum ausgewählten Arbeitsrechner. Da dies bei Inanspruch- nahme relativ langsamer Postleitungen u.U. längere Zeit in Anspruch nimmt, könnte sich inzwischen eine Betriebssituation ergeben haben, die die nunmehr erforderliche Kreation des Sekundärauftrags unmöglich macht, die aber andererseits in Folge des nur quasiaktuellen Zustands der Statusinformation zum Zeitpunkt der Rechnerauswahl am Eingaberechner noch nicht bekannt war. Um Belastungen von Verbundkomponenten (Leitungen, Transportsystem usw.) durch derartige nutzlose Transfers zu vermeiden, erfolgt der Auftragstransfer mehrstufig:

In einem ersten Schritt werden lediglich die den Auftrag beschreibende Information (die sogenannten Auftragskenndaten) zum ausgewählten Arbeitsrechner transferiert. Erst wenn die Kreation des Sekundärauftrags dort gelungen ist, wird ihm die eigentliche Auftragsinformation nachgeschickt. Die in diesem Zusammenhang erforderliche Kommunikation zwischen den beiden KL&KOOP's wird ergänzt durch ein aufwendiges Quittungsspiel mit dem die Informationssicherheit gewährleistet wird.

# 3. Aufgaben des KL&KOOP am Arbeitsrechner

## 3.1 Kreation des Sekundärauftrags

Den Transfer von Aufträgen wickeln, wie schon in 1.2 beschrieben, die beiden KL&KO0P's im Eingaberechner bzw. Arbeitsrechner ab. Bei der Kreation des Sekundärauftrags muss nun zusätzlich berücksichtigt werden, dass die Benutzerverwaltungen in den beteiligten Netzrechnern nicht notwendig identisch sein müssen. KL&KOOP am Arbeitsrechner obliegt deshalb, durch entsprechende Modifikation der Benutzeridentifikation für die Durchlässigkeit der Benutzerverwaltung zu sorgen. Andererseits muss die Rücktransformation möglich bleiben, um die Ergebnisinformation transferierter Aufträge wieder unter der ursprünglichen Benutzeridentifikation etwa am Eingaberechner ausgeben zu können. Eine weitere wesentliche Aufgabe ist die Zuteilung und Verwaltung der Zugriffsberechtigung transferierter Aufträge auf permanente Datenbestände an beliebigen Netzrechnern (Dateiverbund).

## 3.2 Auftragsbegleitinformation

Für jeden transferierten Auftrag wird am Arbeitsrechner Information gehalten, die seine Beziehungen zu den Netzrechnern beschreibt. Es wird dabei im wesentlichen folgendes hinterlegt:

- Der Eingaberechner, um etwa die Ergebnisinformation des Auftrags ohne explizite Benutzerumsteuerung "automatisch" dorthin transferieren zu können
- Die ursprüngliche Benutzeridentifikation vom Eingaberechner
- Die Zugriffsberechtigung auf permanente Datenbestände an beliebigen Lagerrechnern (Dateiverbund)
- Für alle Ausgabeaufträge entweder implizit generierte oder explizit vom Benutzer vorgegebene Angaben über Rechner bzw. Ausgabegerät (Geräteverbund).

An dieser Stelle ist die Kenntnis der Gerätekonfiguration eines adressierten Ausgaberechners notwendig, um etwa fehlerhafte Steuerungen des Benutzers rechtzeitig abblocken zu können.

## 3.3 Transfer der Ergebnisinformation

Alle Ausgabeaufträge gelangen infolge einer speziellen Wahl des Eingabegerätes für transferierte Aufträge mit Unterstützung des Betriebssystems in eine speziell zu diesem Zweck eingerichtete Warteschlange, die sogenannte Sammelwarteschlange. KL&KOOP entnimmt die Aufträge aus der Warteschlange, ordnet ihnen an Hand der Auftragsbegleitinformation den gewünschten Ausgaberechner bzw. das Ausgabegerät zu und übergibt sie dem Transportsystem zum Transfer.

## 4. Aufgaben des KL&KOOP am Ausgaberechner

KL&KOOP ist der Empfänger der Ausgabeaufträge und gibt sie auf das gewünschte Ausgabegerät aus. Auch an dieser Stelle wird durch ein entsprechendes Quittungsspiel mit KL&KOOP am Arbeitsrechner ein Maximum an Informationssicherheit gewährleistet.

## 5. Allgemeine Aufgaben des KL&KOOP

Zu den generellen Aufgaben, die nicht unbedingt einer speziellen KL&KOOP-Funktion an einem bestimmten Rechner zugeordnet werden können, gehören:

- Das Absetzen von Protokollinformation für statistische Auswertung bzw. zum Zweck des Jobaccountings
- Die Auswahl optimaler Lagerrechner bei mehrfach im Netz identisch vorhandenen permanenten Datenbeständen
- Die zeitliche Überwachung aller Transferaktivitäten, um bei eventuellen Störungen des Verbundsystems durch jeweils geeignete Maßnahmen die Informations- bzw. Tranfersicherheit zu gewährleisten
- Das Ablegen wesentlicher Daten wie Zeitüberwachungs- bzw. Auftragsbegleitinformatsion in permanenten Dateien, um im Fehlerfall oder nach Unterbrechungen ohne Informationsverlust darauf wieder aufsetzen zu können
- 0perateurkontakte, um veränderte Betriebssituationen wie etwa den langfristigen Aus fall von Verbundkomponenten oder die Beschränkung von speziellen Betriebsmitteln in den internen Listen berücksichtigen zu können
- Die Bereitstellung von Dienstleistungen auch für Nicht-Transferaufträge im Rahmen des Geräteverbunds, Dateiverbunds und des benutzergesteuerten Datenverbunds.