

**Implementierung eines TR440-
Lastverbund-Systemes unter
besonderer Berücksichtigung
eines Dateiverbundes**

Joachim Backes
Regionales Hochschulrechenzentrum der
Universität Kaiserslautern

Pfaffenbergstraße 95
D-6750 Kaiserslautern

Inhaltsverzeichnis

1 Einführung.....	3
2 Datenfluss im Dateiverbund-Subsystem.....	5
3 Funktion der Komponenten.....	6
3.1 Funktion am Sekundärrechner.....	6
3.2 Funktion am Primärrechner.....	8

1 Einführung

Die Implementierung eines TR440-Lastverbundsystemes am Regionalen Hochschulrechenzentrum der Universität Kaiserslautern¹⁾ erforderte auch die Realisierung eines Subsystemes, welches einem von einem Eingaberechner (*Primärrechner*) zum Arbeitsrechner (*Sekundärrechner*) transferierten Benutzerauftrag die Daten zur Verfügung stellt, die ihm auch am Primärrechner zur Verfügung stehen würden. Unter Daten wollen wir hierbei Dateien der langfristigen Datenhaltung (LFD) verstehen. (Permanente Datenbestände auf Magnetbändern oder privaten Wechselp Plattentürmen wurden in das vorliegende Konzept nicht mit einbezogen).

Der Realisierung lagen folgende Anforderungen bzw. Kriterien zugrunde:

- Reproduzierbarkeit der Betriebsmittel der Benutzeraufträge
- Ausfallsicherheit gegen Systemneustarts
- Effektivität des Datentransfers
- Sicheres Zurückschreiben der Dateien bei Auftragsende
- Beibehaltung der bisherigen Kommandoschnittstelle
- Datenschutz.

Da der Sekundärrechner dem Benutzer im allgemeinen nicht bekannt ist, muss das Zur-Verfügung-Stellen automatisch ablaufen. Daneben darf sich aus Gründen der Transparenz die Benutzerschnittstelle (diese entspricht i.A. den Kommandos zur Dateibearbeitung) nicht ändern. Als Lösung bietet sich an, diese Kommandos auf Kommandoprozeduren abzubilden, die genau die gleichen Spezifikationen besitzen, in die aber zusätzlich eine Instanz eingeschachtelt ist, die den Dateitransfer initialisiert.

Um möglichst alle mit LFD-Dateien verbundenen Leistungen dem Benutzer zugänglich zu machen, werden nicht nur die reinen Datentransport-Kommandos dieser Abbildung unterzogen, sondern auch solche Kommandos, die an LFD-Dateien Änderungen der Verwaltungsdaten vornehmen (Umbenennen, Reservieren, Löschen, Kreieren), oder die den Benutzer über Kenndaten informieren. Das Subsystem (Kurzbezeichnung **DVERB**) besteht aus mehreren Komponenten: den Operatoren KL&DZUG, KL&KODAT, KL&VERDAT sowie verschiedenen Warteschlangen. Relevante Listen des Subsystems werden aus Gründen der Ausfallsicherheit in langfristigen Dateien gehalten.

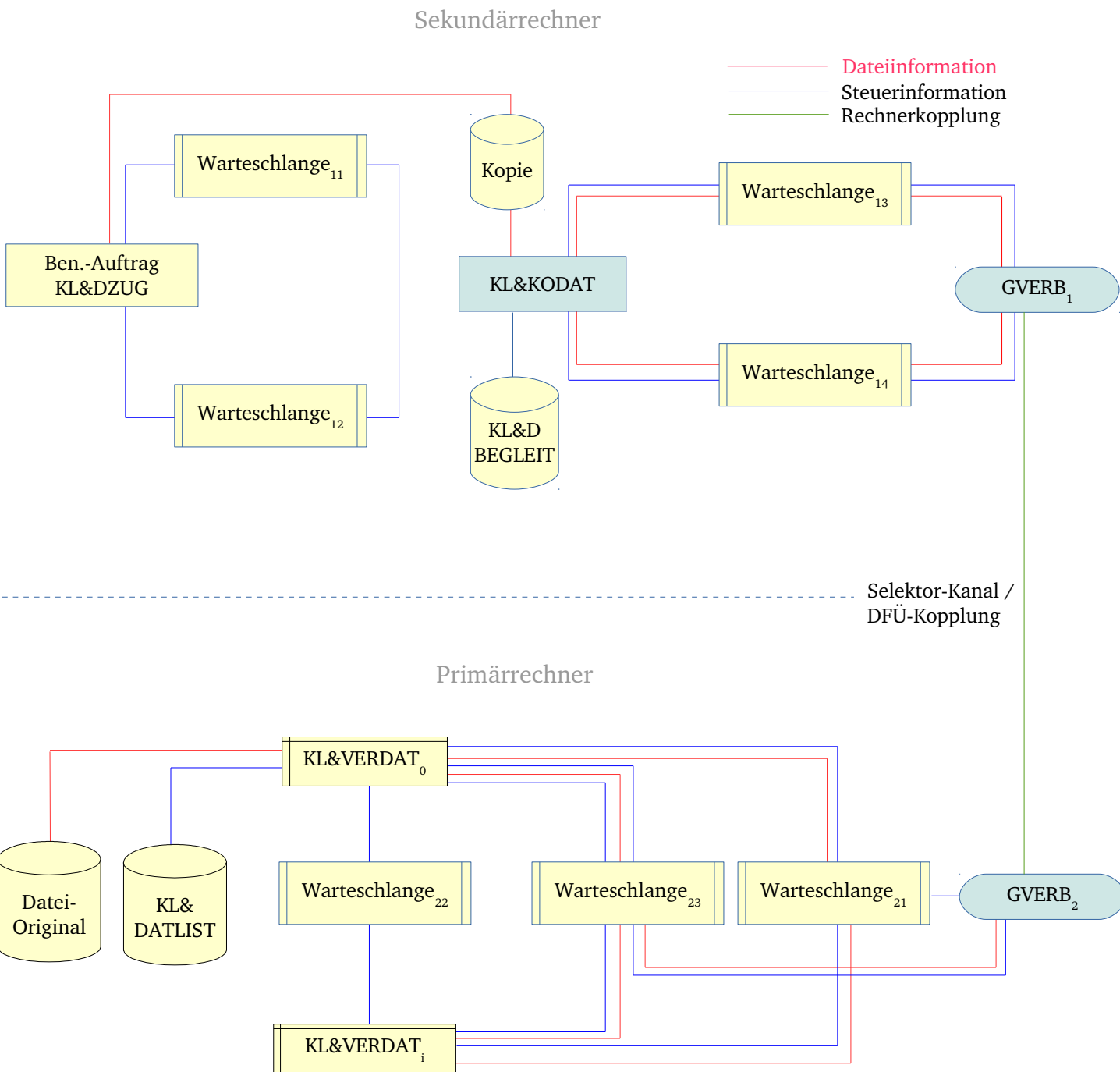
¹ Bezüglich dieses Lastverbundsystemes sei auf den Tagungsband verwiesen (M. Bürkle: Der Kaiserslauterner Lastverbund für TR440-Rechner)

DVERB baut auf Schnittstellen auf, die der Vollständigkeit halber aufgeführt seien:

- 1) **KOMSYS**-Rechner-Verbund (Möglichkeit, von einem primären TR440 Sendungen an Warteschlangen eines über Selektor-Kanal oder DFÜ (TR86S) gekoppelten sekundären TR440 zu übertragen) ↗ *Gerätesendung*
- 2) Gebiets-Verbund **KOMVERB** (Möglichkeit, Gebiete an Benutzerauftrag in gekoppelte TR440 zu übertragen) ↗ *Gebietssendung*
- 3) **DVERB**

Die System- bzw. Hardwarenähe nimmt von 1) nach 3) ab.

2 Datenfluss im Dateiverbund-Subsystem



3 Funktion der Komponenten

3.1 Funktion am Sekundärrechner

Wie bereits erwähnt, werden die LFD-Dateibearbeitungskommandos auf Kommandoprozeduren abgebildet mit dem Ziel, den Operator KL&DZUG zu starten, der den Kontakt mit DVERB einleitet. KL&DZUG übermittelt seine Steuerinformation, die ja Einzelheiten über den geforderten Bearbeitungsmodus enthält, sowie einen Teil der Auftragskenndaten über eine Warteschlange an den Operator KL&KODAT. KL&KODAT wiederum überprüft zunächst die Auftragskenndaten auf Konsistenz, um illegale Datentransporte zu verhindern. Sollte ein derartiger Fall auftreten, wird unter Zuhilfenahme einer Quittungswarteschlange die Benutzeranforderung negativ quittiert. Ansonsten ermittelt KL&KODAT aus den in der Steuerinformation aufgeführten Dateien diejenigen, die Verbundleistungen erfordern.

Für die lokal verfügbaren Dateibestände erzeugt KL&KODAT aus der empfangenen Kontrollinformation ein leicht modifiziertes Dateikommando und sendet dieses im Klartext an den Benutzerauftrag zurück. Dort wird dieses Kommando (erneut) ausgeführt (Technik: Entschlüsslerstart in der 2. Stufe).

Für die nicht lokal verfügbaren Dateien dagegen werden Verbundleistungen benötigt. Zu diesem Zweck wird die im Kommando geforderte Aktion in kompakter Form an KL&VERDAT als Empfänger übertragen. Im einzelnen kann es sich darum handeln,

1. LFD-Dateien vom Primär- zum Sekundärrechner zu transferieren,
2. LFD-Dateien vom Sekundär- zum Primärrechner zu transferieren,
3. Dateikenndaten am Primärrechner zu verändern,
4. Dateikenndaten auszuliefern

Der wichtigste Fall (1.) sei näher erläutert: mittels der KOMVERB-Schnittstelle beauftragt KL&KODAT den primärrechnerseitigen KL&VERDAT, eine LFD-Datei zum Sekundärrechner zu übertragen. Nach erfolgtem Transfer steht dort die angeforderte Datei in Form von Gebieten zur Verfügung, aus denen KL&KODAT eine Scratch-Datei erzeugt. Diese Scratch-Datei wird auf einen "mittelfristigen" Träger wie z.B. Wechselpatte im Vielfachzugriff kopiert, und zwar unter der gleichen Katalogbezeichnung wie am Primärrechner mit einem Hilfspasswort. Um diese Dateikopie dem Benutzer zugänglich zu machen, erhält KL&DZUG als Quittung zum Transfer eine Sendung, welche Kommandos im Klartext zum Bereitstellen der Datei enthält: Einschleusen, Verlagern auf Platte oder Trommel o.ä.. Eine von einem anderen Benutzer bereits "eingeschleuste" Datei braucht natürlich nicht nochmals zum Sekundärrechner transferiert zu werden. In den meisten Fällen reicht es aus, dem Benutzer eine Kommandofolge zurückzusenden, die das Einschleusen der Datei vom Hilfsträger bewerkstelligt. Umgekehrt darf eine Datei, wenn sie ausgeschleust werden soll, erst dann auf dem Hilfsträger gelöscht werden, wenn kein Auftrag mehr darauf zugreift.

Falls in diesem Zusammenhang eine Datei vom Sekundär- zum Primärrechner zu transferieren ist, sei es spontan oder auch zum "Ausschleusen" (entweder auf Anforderung hin oder falls der Benutzerauftrag sich ohne explizites Ausschleusen beendete), wird der Transfer durch KL&KODAT zunächst angekündigt. Danach wird der Benutzer, falls erforderlich, veranlasst, KL&KODAT die zu transferierenden Dateien zur Verfügung zu stellen. Erst dann kann der eigentliche Transfer beginnen.

Über den gesamten Transfer wird in der Logdatei KL&DBEGLEIT Buch geführt. Um nach Operator- oder Systemneustarts wieder aufsetzen zu können, wird diese Information in der LFD abgelegt.

Bei Informationsdiensten über LFD-Dateien schließlich werden die durch KL&VERDAT festgestellten Kenndaten in kompakter Form an KL&KODAT übertragen, der sie, in Klartext umgeformt, an KL&DZUG weiterreicht; dieser wiederum sorgt für eine Ausgabe des Textes auf Drucker oder Terminal.

Anmerkungen zur RZ-Organisation: Damit KL&KODAT entscheiden kann, ob ein Dateitransfer erforderlich ist oder nicht, muss er wissen, welche LFD-Kataloge an welchem Rechner des Netzes lokalisiert sind. Dies erfordert einmal, dass die Katalogbezeichnungen bis auf einige wohldefinierte im Netz eindeutig sind, zum andern, dass Katalog-Änderungen an irgendeiner Stelle im Netz unverzüglich jedem beteiligten Netzrechner mitzuteilen sind.

3.2 Funktion am Primärrechner

Die Leistungen des Dateiverbund-Subsystems werden am Primärrechner vollständig durch KL&VERDAT abgedeckt. Zu diesen zählt einmal das Handling der Dateiinhalte, d.h. aufbereiten der zu sendenden Dateien, um sie GVERB übergeben zu können, oder Rückumwandlung der vom Sekundärrechner erhaltenen Gebiete in Dateien sowie das Übertragen von Kenndaten oder Quittungen zur Kenndatenänderung an den Sekundärrechner.

Wichtiger aber ist die Simulation der LFD-Dateizugriffskoordination, eines Dienstes, der vom Betriebssystem lokal garantiert wird. Da aber Dateien von verschiedenen Rechnern her *ingeschleust* sein können, muss Sorge dafür getragen werden, dass bei *schreibenden* bzw. *lesenden* Zugriffen auch von anderen Rechnern aus nicht mehr *lesend* bzw. *schreibend* zugegriffen werden kann. Das hat zur Folge:

- Jede zu einem Sekundärrechner aufgrund von Einschleusungsanforderungen transferierende Datei muss lokal im gleichen Modus eingeschleust werden;
- für jeden externen Einschleusevorgang müssen entsprechende Zugriffszähler geführt werden;
- Dateioriginale dürfen erst dann ausgeschleust werden, wenn die Zugriffszähler normiert sind.

Insbesondere heißt dies, dass u.U. mehrere KL&VERDAT-Aufträge im Primärrechner aktiv sein müssen, da das BS3 je Auftrag nur eine begrenzte Anzahl angemeldeter Dateien zulässt. Generiert werden diese verschiedenen KL&VERDAT-Jobs zentral durch einen "Ur"- KL&VERDAT (KL&VERDAT₀) bei Bedarf oder auf Anforderung hin durch solche Nachfolger, deren Limit erreicht ist. KL&VERDAT₀ sorgt auch dafür, dass bei Bedarf die Anzahl reduziert wird oder bei Ausfall eines Nachfolgers dieser regeneriert wird.

Die einzelnen KL&VERDATs kommunizieren über Warteschlangen. In einer zentralen LFD-Datei (aus Gründen der Ausfallsicherheit und Effizienz bei System-Neustart) führen die KL&VERDAT-Operatoren Buch über die verschiedenen Datei-Anmeldungen. Dazu dient die Datei KL&DAT-LIST, bestehend aus den drei Teillisten

- **SOHNLIST:** verwaltet durch KL&VERDAT₀, enthält Information über die Nachfolger-KL&VERDATs.
- **ANLIST:** ein Listenelement, das für jede Datei angelegt wird, enthält die Summe der Rechner und Aufträge, die diese Datei *ingeschleust* haben, sowie das Kennzeichen desjenigen KL&VERDAT, der die Datei verwaltet.
- **AUFTRAGSLISTE:** zu jedem KL&VERDAT existiert genau ein Listenelement, das sämtliche von diesem KL&VERDAT *ingeschleusten* Dateien enthält.

Die in KL&DATLIST offenbar enthaltene Redundanz dient einer effektiveren Restart-Behandlung.

Bei der Benutzerverwaltung ist zu hinterlegen, dass die KL&VERDAT-Operatoren auf **sämtliche Dateien aller LFD-Kataloge** zugreifen dürfen. Die Passwortfrage bleibt davon natürlich unberührt; Passwörter sind immer durch das Betriebssystem abzufragen, um einen nicht berechtigten Benutzer auszuschließen. Außerdem ist naturgemäß der Operator KL&VERDAT gegen illegale Starts durch Benutzer gesperrt.