

Datenbanken in Realzeitumgebung am Beispiel eines Betriebsleitrechners.

Dipl.-Math. Benno Schneiders

1. ABSTRACT

Am Beispiel eines auf einem Doppelrechnersystem realisierten Betriebsleitrechners wird aufgezeigt, daß ein Datenbanksystem in einer solchen Systemumgebung sowohl sehr zeitkritische, realzeit-orientierte Aufgaben zu erfüllen hat, als auch den Aufgabenbereich eines normalen Informationssystems abdecken muß. Dazu wird zunächst die allgemeine Aufgabenstellung erläutert, ein Hard- und Softwaremodell vorgestellt und dann die speziellen Anforderungen an das Datenbanksystem näher erläutert.

Am Beispiel des Datenbanksystems PISA/DB wird aufgezeigt, wie durch spezielle Datenstrukturen und Funktionen die Forderungen nach Geschwindigkeit, Ausfallsicherheit und Verfügbarkeit erfüllt werden können. Dazu gehören z.B. die Möglichkeit des Doppelschreibens, eine komfortable Pufferverwaltung, Dateien mit direktem Zugriff und zyklische Dateien.

Zum Schluß wird dargestellt, wie auf einem Doppelrechnersystem durch einfache Maßnahmen der Wiederanlauf auf dem Standby-Rechner zu gewährleisten ist.

2. AUFGABENSTELLUNG

Aus der immer weiter fortschreitenden Automatisierung in den Produktionsbetrieben, ergibt sich die Anforderung, zwischen der schon vorhandenen

Groß-EDV und der Prozeßsteuerung eine dritte Rechner-Ebene einzuschleiben, den sogenannten Betriebsleitrechner (BLR). Generell kann man die Aufgabe des BLR darin sehen, beliebige Prozessvariable abzufragen, die Informationen aufzubereiten, abzuspeichern und an das Management weiterzureichen.

Im Rahmen dieses Referates, wird davon ausgegangen, daß der BLR in einem Produktionsbetrieb steht und folgende fünf Aufgabenkomplexe zu bewältigen hat.

1) Die Produktionsüberwachung

Alle relevanten Daten der laufenden Produktionen (Zählerstände, Meßwerte, Zustände, Störmeldungen) werden automatisch gesammelt und abgespeichert. Störungen und Produktionsabweichungen werden über Bildschirme an das Management gemeldet. Sämtliche angefallene Daten zu einer Produktion werden über einen bestimmten Zeitraum aufbewahrt.

2) Statistik

Die aus der Produktionsüberwachung anfallenden Daten werden nach Abschluß eines Produktes automatisch verdichtet und für spätere statistische Auswertungen abgespeichert. Diese Daten werden über einen längeren Zeitraum aufbewahrt. Extrakte aus den Daten werden an die Groß-EDV übertragen.

3) Disposition

Die Disposition für die anstehenden Produktionen werden aufgrund von vorhandenen Stammdaten und standardisierten Dispositionen den aktuellen Bedürfnissen angepaßt. Die Maschinen- bzw. Produktionsleiter rufen aus den erstellten Dispositionen ihre anstehenden Aufgaben ab.

4) Stammdatenverwaltung

Hier werden die Informationen über den Maschinenpark und die Produktionspfade verwaltet und gespeichert.

5) Auswertungen

Das Betriebsmanagement hat die Möglichkeit, sich anhand der Produktionsdaten über den Zustand der laufenden Produktionen zu informieren, bzw. anhand der statistischen Daten-Aussagen über Betriebsabläufe bzw. über Schwachstellen im Betrieb zu verschaffen.

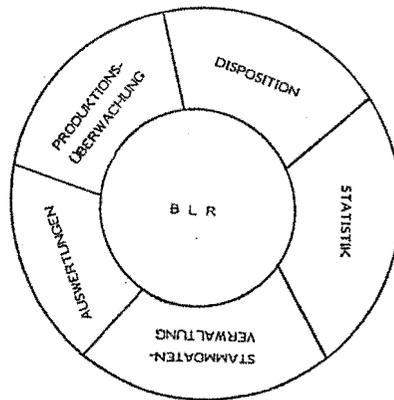


Abb. 1: Aufgaben des Betriebsleiters

Man sieht, daß die Aufgabenstellungen an den Betriebsleiters völlig unterschiedliche Anforderungen stellen. Es müssen Prozessdaten automatisch erfaßt werden. Dazu wird eine schnelle Reaktion und eine hohe Ausfallsicherheit von dem System gefordert. Andererseits dient das System als Informationssystem, d.h., die erfaßten Daten müssen so flexibel strukturiert sein, daß alle denkbaren Auswertungen in akzeptablen Zeiten durchführbar sind.

Diese komplexen Anforderungen sind natürlich nur mit einer leistungsfähigen Hard- und Software zu erfüllen.

3. HARDWARE-KONZEPT

Die Auslegung der Hardware hängt von den Anforderungen an die Ausfallsicherheit des Gesamtsystemes ab. Je sicherer das System sein soll, um so größer werden die Hardwarekosten. Allerdings sollten bei der Kostenkalkulation berücksichtigt werden, welche Unkosten durch die Einschränkung oder sogar Einstellung der Produktion entstehen können, wenn der BLR einmal ausfällt.

Die optimale Auslegung der Hardware eines BLR könnte in etwa wie folgt aussehen:

- mindestens zwei Frontendrechner als Datenkonzentratoren zwischen den Prozessen und dem BLR. Diese sollten über eine beschränkte Plattenkapazität verfügen, damit bei einem eventuellen Ausfall des BLR, bzw. bei einem Stau vor dem BLR, die anfallenden Daten zwischengespeichert werden können.
- Der BLR wird als Doppelrechnersystem ausgelegt. Dabei dient ein Rechner nur als Standby und wird im Normalfall höchstens für Entwicklungs- und Testarbeiten genutzt.
- Die gesamte Peripherie ist umschaltbar von einem Rechner zum anderen.
- Die Plattenkapazität muß so ausgelegt sein, daß alle Daten doppelt geführt werden können. Auf diese Weise ist man auch gegen Plattenfehler geschützt.

Abbildung 2 stellt einen solchen optimalen BLR dar. Abhängig von der jeweiligen Anwendung können natürlich einige Komponenten einfach ausgelegt werden bzw. ganz entfallen (z.B. die Frontendrechner).

4. SOFTWAREKOMPONENTEN

In Abbildung 3 ist der generelle Aufbau des Softwaresystems auf dem betrachteten BLR dargestellt. Im Prinzip besteht das Gesamtsystem aus drei Ebenen:

- dem Betriebssystem,
- der Standardsoftware, dazu gehören:
 - + Dialogsprache
 - + Reportgenerator

- + Datenbanksystem
- + DFÜ-Software
- der Anwendersoftware
hier also:
 - + Stammdatenverwaltung
 - + Disposition
 - + Statistik
 - + Produktionsüberwachung
 - + spezielle Auswertungen z.B. Darstellung von Prozesszuständen an den Bildschirmen.

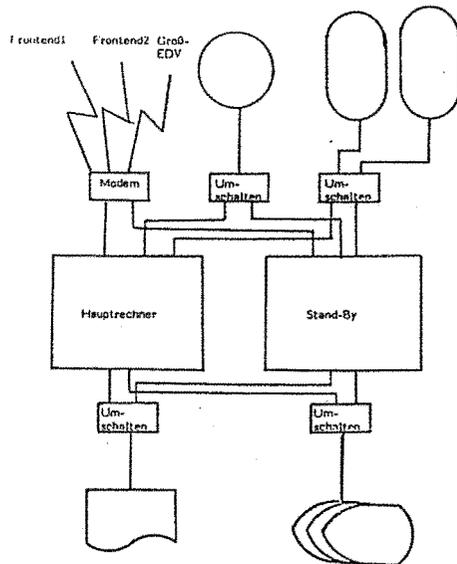


Abb. 2: Hardware eines BLR

Im Folgenden sollen die Aufgaben des Datenbanksystemes in dieser Umgebung näher betrachtet werden. Dabei geht es weniger um solche 'natürlichen' Aufgaben eines Datenbanksystemes wie z.B. die Stammdatenverwaltung bzw. die Auswertung von Daten in Listen und Formularen, sondern mehr um die besonderen Anforderungen an das DB-System, die sich aus dem Aufgabengebiet der Produktionsüberwachung ergeben.

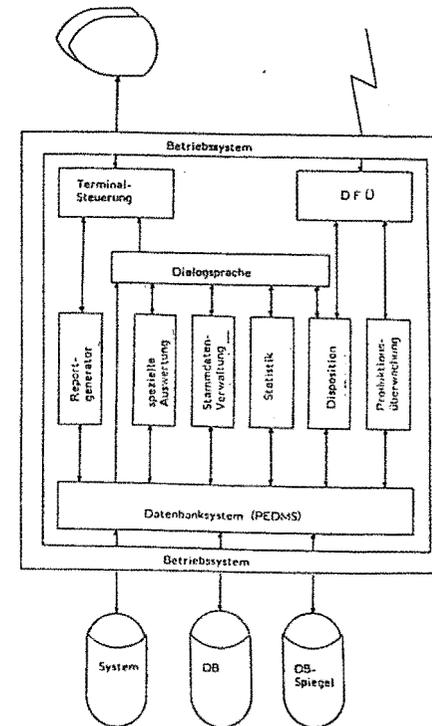


Abb. 3: Blockschaltbild der Software des BLR

5. AUFGABEN DES DATENBANKSYSTEMES

Zunächst soll kurz erläutert werden, was hier unter Produktionsüberwachung verstanden werden soll.

Der Frontendrechner fragt periodisch alle Zähler und Meßwerte der z.Z. laufenden Produktionen ab, sammelt sie zu Telegrammen und schickt diese an den BLR. Darüberhinaus empfängt er ereignisgesteuert Zustandsänderungen und Störmeldungen und gibt diese in gesonderten Telegrammen direkt an den BLR weiter.

Auf dem BLR werden die unterschiedlichen Telegramme an die Verarbeitungs-

routinen verteilt. Zunächst muß geprüft werden, ob die Meldungen nicht schon einmal empfangen wurden und eventuell schon verarbeitet sind. Dies kann z.B. bei einem Verarbeitungstau auf dem BLR passieren -die Quittung kam nicht in einem definierten Zeitraum-, bzw. während der Synchronisation zwischen BLR und Frontend nach einem Systemausfall.

Jedes Telegramm, daß im Normalfall n-Meldungen enthält, wird in einer Datenbanktransaktion abgearbeitet. Dadurch reduziert sich der Synchronisationsaufwand zwischen BLR und Frontend auf Telegrammebene.

Wurde das Telegramm als korrekt erkannt, wird jede einzelne Meldung in der Datenbank abgespeichert und das Telegramm dem Frontendrechner quittiert.

5.1 Datenstrukturen zur Produktionsüberwachung

Um die oben erläuterten Aufgaben wahrnehmen zu können, werden im Rahmen der Produktionsüberwachung zwei unterschiedliche Datenstrukturen zur Beschreibung und Speicherung der Prozessdaten definiert:

- Statusdatei

Die Datei spiegelt zu jedem Zeitpunkt ein aktuelles Abbild der laufenden Prozesse wieder, d.h. zu jeder Prozessvariablen wird der aktuelle Zustand und der Zeitpunkt der letzten Aktualisierung nachgehalten.

Diese Dateien sind normalerweise nach Maschinen strukturiert und enthalten pro Prozessvariable folgende Informationen:

- + letzte Aktualisierungszeit
- + Zustand aktuell
- + Zustand normal
- + Aktionscode
- + Sonstiges (z.B. Toleranzbereiche).

Dabei gibt der Aktionscode Auskunft darüber, was bei Änderungen von Zuständen für Sonderaktionen zu erfolgen haben.

Die Datei wird für folgende Aufgaben benutzt:

- + zur Überprüfung, ob eingehende Meldungen schon verarbeitet wurden,
- + zur Darstellung des aktuellen Prozessabbildes auf dem Bildschirm.

- Produktionsdatei

In dieser Datei werden alle Zustandsänderungen, Zähler- und Meßwerte in ihrem zeitlichen Ablauf festgehalten. Das heißt pro Signal/Prozessvariable werden bei jeder sie betreffenden Meldung folgende Informationen abgespeichert:

- + Signalnummer
- + Zeit
- + Zustand.

Diese Datei ist produktionsorientiert, d.h. zeitliche Auswertungen über bestimmte Produktionen sind möglich.

5.2 Datenstrukturen und Datenbankfunktionen in PISA/DB

In den Statusdateien spiegelt sich der Aufbau des Produktionsbetriebes wieder, d.h. ihre Struktur wird sich kaum ändern. Deshalb empfiehlt es sich hier mit festen Rekordlängen zu arbeiten. Man definiert unterschiedliche Rekordtypen, z.B. die Zähler pro Maschine ergeben einen Rekordtyp.

Jeder Rekordtyp hat den gleichen formalen Aufbau:

- Maschinenummer,
- Information zu Signal Nr. 1,
- Information zu Signal Nr. n.

Der Zugriff auf diese Dateien geschieht über einen Cala-Algorithmus. Dieser Algorithmus stellt sicher, daß der Zugriff auf einen Rekord maximal einen E/A erfordert.

Da in PISA/DB durch einen komfortablen Pufferungsmechanismus die Möglichkeit besteht, speicherresidente Dateien einzurichten, bedeutet dies, ein lesender Zugriff wird immer ohne E/A erfolgen, während ein ändernder Zugriff genau einen E/A benötigt.

Im Gegensatz zu den Statusdateien hat die Produktionsdatei einen sehr variablen Aufbau. Es ist nicht vorherbestimmbar, wieviel Daten zu einer Produktion anfallen werden, andererseits sollen die zeitlichen Abläufe einer Produktion auswertbar sein, d.h. jeder Satz in der Produktionsdatei muß durch eine Produktionsnummer eindeutig einer Produktion zuzuordnen sein. Auch hier wird so verfahren, daß unterschiedliche Dateien für Zähler, Meßwerte und Zustände eingerichtet werden, aber nicht jede Änderung einer Prozessvariablen ergibt einen neuen Satz in der Produktionsdatei. Vielmehr werden hier die Möglich-

keiten der multiplen Felder in PISA/DB ausgenutzt.

Der generelle Satzaufbau sieht wie folgt aus:

- Produktionsnummer,
- Information zu Signal 1
- .
- .
- Information zu Signal n.

Dabei besteht die Information zu jedem Signal aus einer Gruppe von multiplen Feldern. Beim Start einer Produktion, wird ein Satz dieses Typs mit den Ausgangswerten pro Signal abgelegt. Bei jeder Meldung zu einem Signal, wird zu jedem Feld der entsprechenden Gruppe ein Wert hinzugefügt. Erst wenn der Satz eine vorher definierte Größe überschritten hat, wird zur gleichen Produktion ein neuer Satz angelegt.

Für die Produktionsdateien wird eine weitere spezielle Funktion von PISA/DB angewendet, die sogenannten zyklischen Dateien. In den Schemadefinitionen für diese Art von Dateien kann eine Zykluszeit vorgegeben werden, d.h. alle Daten, die in dieser Datei abgelegt werden, werden während dieser Zykluszeit aufbewahrt und anschließend automatisch vom Datenbanksystem gelöscht. Der freiwerdende Platz wird sofort wieder für neue Produktionsdaten frei.

Auch für die Produktionsdaten wird die komfortable Pufferverwaltung von PISA/DB ausgenutzt. Zwar können die Dateien wegen ihrer Größe nicht vollständig speicherresident gehalten werden, aber man stellt den Produktionsdateien exklusiv einen genügend großen Pufferpool zur Verfügung. Dadurch ist einerseits sichergestellt, daß durch andere Arbeiten mit dem Datenbanksystem keine Produktionsdaten aus dem Pufferpool gealtert werden, andererseits werden die aktuell bearbeiteten Sätze ständig speicherresident sein.

Ein weiteres wesentliches Kriterium, welches PISA/DB dazu prädestiniert in solchen Anwendungsumgebungen eingesetzt zu werden, ist die völlige Reorganisationsfreiheit der Daten, die automatische Wiederverwendung von freiwerdendem Speicherplatz und die Möglichkeit zusätzlicher Datenstrukturen parallel zur laufenden Anwendung zu definieren. Dadurch gewährleistet das Datenbanksystem einen unterbrechungsfreien 24-Stunden-Betrieb.

5.3 Datensicherheit in PISA/DB

PISA/DB bietet mehrere unterschiedliche Sicherungsverfahren an. Diese können sowohl einzeln als auch in Kombination implementiert werden. Als eine sinnvolle Kombination hat sich erwiesen:

- das Transaktion-Undo
Dieses beinhaltet sowohl das Backout Transaktion (BOT) im laufenden Betrieb, als auch das Zurücksetzen offener Transaktionen nach einem Systemausfall,
- das Doppelschreiben
Alle Änderungen der Daten werden vom Datenbanksystem automatisch auf zwei Platten mitgeführt. Beim Ausfall einer Platte arbeitet das System mit der verbleibenden Platte weiter. Der Anwender hat die Möglichkeit einen Drive aus dem System auszukoppeln -Erzeugen einer Kopie-, und eine frische Platte in den Drive einzulegen. Das System zieht die neue Platte automatisch auf den aktuellen Stand hoch.

5.4 Wiederanlauf

Unter Wiederanlauf soll hier der Ausfall des Hauptrechners und die Übernahme der Arbeit durch den Stand By verstanden werden. Das Datenbanksystem schreibt alle seine Restartinformationen auf die Platte. Beim Umschalten auf den Stand-By-Rechner kann das dortige Datenbanksystem also einen normalen Restart durchführen (d.h. alle offenen Transaktionen werden zurückgesetzt).

Die Anwendung, die ihre Restartinformationen auch auf der Platte vorfindet, erhält vom Datenbanksystem die Information, welche Transaktionen zurückgesetzt werden und kann sich schnell mit dem Frontendrechner synchronisieren (zur Erinnerung: ein Telegramm entspricht einer Transaktion).

Schneiders, Benno
Rhonestr. 2, 5000 Köln 71
0221 70 91 233