

## Tuningfall-3: Umfassendes DB2-Tuning bei einem Genossenschaftsverband in Norditalien

Der *Nutzen eines Datenbanksystems* entsteht erst aus der *intensiven Nutzung der verfügbaren Information* und der daraus resultierenden Wertschöpfung - das aber zu *Kosten, die in Relation zum erzielten Vorteil* durch die gewonnenen Information stehen.

Die Konfigurationsmöglichkeiten in DB2 sind so angelegt, dass alle erforderlichen Prozesse und Anwendungen aufs Effizienteste bedient werden können. Dazu müssen nicht nur alle (System-) Parameter passend gewählt, sondern auch die Datenstrukturen optimal in die physische Umgebung umgesetzt sein. Insbesondere gilt es aber *Applikationen* architektonisch so abzubilden und Programme so zu schreiben, dass Performanceziele erreicht werden und ein Minimum an Kosten entsteht.

Den *größten Effekt in der Erhöhung der Performance* erreicht man, wenn man Applikationen nach auffälligen SQL-Statements durchsucht.

### Aufgabenstellung

In der **folgenden Analyse wurden definierte Systeme (online und Batch)** eines Raiffeisen Genossenschaftsverbands in Südtirol untersucht. Ziel war es die DB2 Performance signifikant zu erhöhen und die CPU-Kosten drastisch zu senken.

Diese **Untersuchung** sollte sich auf Pläne und Packages, in bestimmten, dezidierten CICSUmgebungen, die vom Kunden vorgegeben worden waren beschränken und aufgrund von Systemmessungen über Omegamon für DB2 (später MainView for DB2) als kritisch evaluiert worden waren. Dies definierte auch das Ziel der vorliegenden Aufgabe:

- Analyse diverser **DB2-Programme/DB2-Packages**
- Erkennen der Auffälligkeiten und kritischen Verhaltensweisen (CPU-Verbrauch, "elapsed time"-Verbrauch)
- Erarbeiten und Planen der Lösungsmaßnahmen zur **Erhöhung der DB2 Performance**
- Erfolgskontrolle der eingeleiteten Performance-Maßnahmen, Vergleich der Ergebnisse und Vorschlagen weiterführender Maßnahmen

Nicht Bestandteil der vorgegebenen Aufgabe war:

- Implementieren und Testen der vorgeschlagenen Lösungen (wurde vom Kunden durchgeführt)

### Vorgehen und Ablauf

Das Verhalten der Programme/Module/Packages wurde mit Omegamon for DB2 ermittelt. In einem nächsten Schritt wurden die über die Messung erkennbaren Auffälligkeiten weiter detailliert und konnten dann genau analysiert und korrigiert werden.

Die meisten *Auffälligkeiten* konnten über *Umschreiben der SQL-Queries und die Neudefinition zusätzlicher Indizes* behoben werden. Dabei waren die Änderungsvorschläge optional und rentierten sich nur bei hoher Nutzungsfrequenz und entsprechendem (CPU-) Einsparpotential. Dies wurde in Einzelfällen von der DB2 DBA zusammen mit der SKC GmbH in gesonderten Lauftests untersucht.

Offiziell sollte es *keine dynamischen Zugriffe* auf DB2-Ressourcen geben. Messungen zeigten allerdings, dass sehr wohl dynamische Aktivitäten die DB2-Performance beeinträchtigten. Es fiel, wie zu erwarten, auf, dass die *dynamisch auf das DB2 durchgeführten Zugriffe*, ein Vielfaches eines statischen SQLZugriffs an DB2 Ressourcen verschlangen. Ein stabiler Faktor lag hier beim *3-4 fachen des CPUVerbrauchs verglichen mit statischem DB2-SQL*.

Die Analyse brachte noch **Tuningpotentiale in anderen (nicht direkt untersuchten) Programmen / Modulen** zu Tage. Sie bewegten sich jedoch im „**elapsed time**“ Verbrauch und in der Fähigkeit der parallelen Abwicklung bestimmter Applikationen und schlugen so nicht direkt auf (MIPS-) Accountingkosten durch. Die Aufhebung dieser „Blockaden“ im DBMS brachte **Vorteile für Durchsatz- und Antwortzeiten** des gesamten Applikationskomplexes.

Da – wie von der System- und DB2 DBA bemerkt – übermäßig **häufig SORTs durchgeführt** wurden, wurde auch geprüft, ob die SORTPOOLSIZE groß genug gehalten und andere betroffene Systemparameter für die DB2 Performance optimal eingestellt waren.

Weitere Verbesserungen in der DB2 Performance und Reaktionszeit der untersuchten Systeme wurden über eine **"exception analysis"** auf die existierende DB2-Produktionsumgebung sichtbar.

Die aus der Analyse resultierenden **DB2-orientierten Änderungen** und die vorgeschlagenen **systemtechnischen und systemnahen Maßnahmen** wurden von der verantwortlichen DB2 DBA Abteilung in der zentralen Systembetreuung unmittelbar in Angriff genommen.

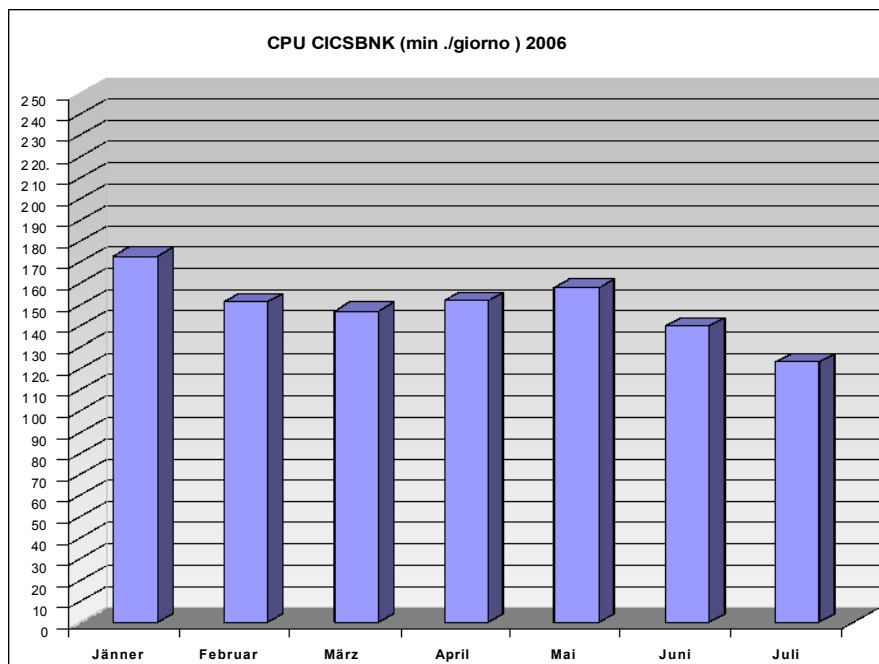
Ein Problem der mittelmäßigen DB2 Performance lag auch in der Anwendungsarchitektur: Das Thema war "Kapselung der fachlichen Funktionen". Zukünftig sollte überlegt werden, ob die Anwendungsarchitektur dahingehend erweitert wird, dass bestimmte **zentrale Routinen die Möglichkeiten von DB2 voll ausschöpfen** sollten und damit „**stored procedures**“, „**triggers**“ und ähnliche Objekte ins Anwendungsentwicklungsportfolio mitaufgenommen werden konnten.

**Das implementierte physische Datenmodell**, soweit dies aufgrund der betrachteten Ausschnitte beurteilt werden konnte, war stabil und optimiert.

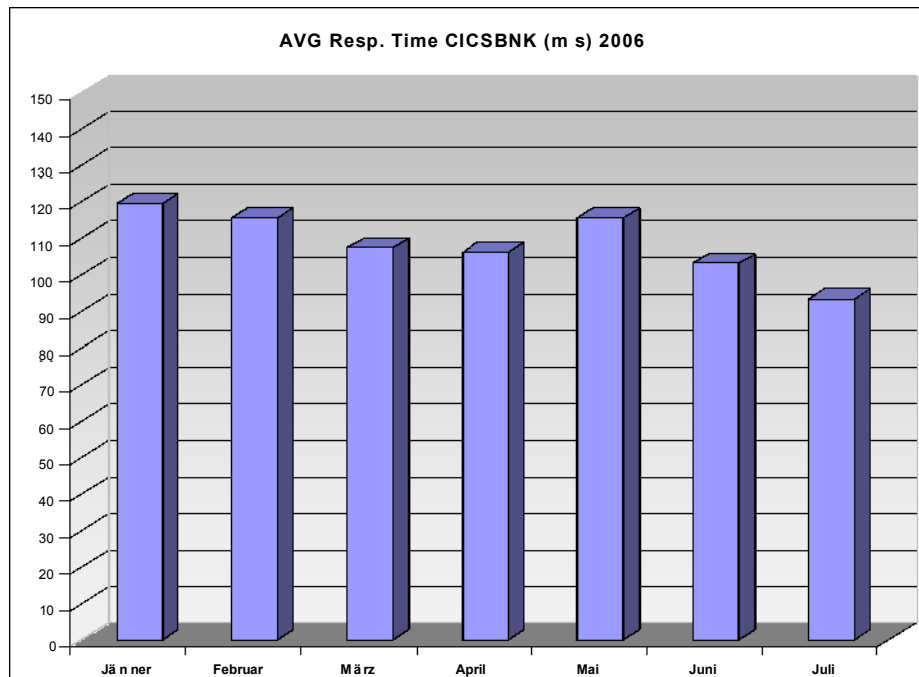
## Ergebnisse

Die gewonnenen Erkenntnisse wurden im Rahmen von "workshops" und Empfehlungen an die DB2-Anwendungsentwickler kommuniziert und weitergegeben.

Die Anwendungspakete wurden CICS-orientiert – also pro CICS - untersucht, da sie so betrieben wurden.



Zuletzt wurde am 9.8.2006 von der Systemabteilung eine *durchschnittliche Ersparnis an CPU-Zeit im ersten untersuchten CICS (CICSBNK) von ca. 18%* bestätigt.



Dabei war zu beachten dass der Januar wegen der **Jahresabschlüsse** natürlich eine höhere Belastung aufweisen musste, als alle anderen Monate im Jahr!

Die Arbeit von DB2 Performance Optimierung und Tuning begann mit den Analysen im April und zeigte im Juni erste Wirkungen.

Auch die Antwortzeiten hatten sich entsprechend verbessert: Insgesamt ergab sich **für das gesamte System** mit diesem ersten Schritt eine gesamte CPU-Zeitersparnis von ca. **2,5%**.

**"Quantitativamente il risparmio sulla CPU complessiva e' stato pari al 2,5%, che a me sembra un buon risultato.... "**

(Maurizio Piubelli, Leiter Systemadministration)

Das Projekt wurde mit der Untersuchung und Verbesserung weiterer CICS (CICSAFP, CICS MOP, CICSVER...) fortgeführt.