Das Datenbank-Konzept von dbGonzales:

- Besseres Design und leichtere Abbildung Ihrer Datenmodelle durch echte Objektorientierte- Datenhaltung.
- Migration von bestehenden Anwendungen durch Unterstützung von relationalen Datenmodellen.
- Verteilte Datenhaltung, Lastenverteilung, erhöhte Ausfallsicherheit und echte Skalierbarkeit durch **Smart-Client** / **Small-Server**.
- Höchste Geschwindigkeit im Netz durch **Warp-I/O**.
- Konsistente Daten durch Multi-Generationen-Transaktionen.
- Hoch optimierte Datenbank-Engine.
- Non-Blocking-Read und Write.
- Zero-Overhead Commit und Rollback.
- Minimaler Overhead - Codegröße um 400KB.
- Verfügbar für Windows und Linux

Einleitung – oder was ist dbGonzales ?

Diese Frage ist nicht in einem Satz zu beantworten. Deshalb möchten wir im folgenden Absatz einen kurzen Überblick über die Konzeption von ***dbGonzales*** geben.

Die Grundkonzeption von ***dbGonzales*** unterscheidet sich in vielen Dingen von den Konzepten einer klassischen relationalen Client/Server Datenbank.

So arbeitet ***dbGonzales*** als **Smart-Client** / **Small-Server** System. Darunter ist zu verstehen, dass jeder Client seine Abfragen selbstständig durchführt. Im Gegensatz zu einer lokalen Datenbank wie dBase, Paradox oder Access, werden jedoch alle Schreibzugriffe von dem jeweiligen ***dbGonzales*** Server durchgeführt. Somit verhält sich ein Client im Prinzip wie ein Knoten in einem Datenbank-Cluster.

Auf den ersten Blick ist der Vorteil dieser Vorgehensweise wahrscheinlich nicht klar ersichtlich, da nach gängiger Meinung die Daten von einem lokalen Massenspeicher (Festplatte oder RAID-System) schneller als über das Netz gelesen werden können. Wir haben für ***dbGonzales*** zur Vermeidung dieses Nadelöhrs eine spezielle Technik entwickelt, die wir **Warp-I/O** nennen. Auch wenn eine größere Datenbank mit mehreren Millionen Datensätzen über das Netz gelesen werden muss, erledigt ein ***dbGonzales***-Client mit modernem GHz Prozessor diese Aufgabe, dank **Warp-I/O** genauso schnell, wie wenn die Daten von einer sehr schnellen Festplatte gelesen würden.

Veith System GmbH . Laiming 3 . 83112 Frasdorf . Germany
Tel: 08052-2636 Email: info@veith-system.de

Zur weiteren Geschwindigkeitssteigerung synchronisiert der Server seinen Datenbestand mit dem Clients und überträgt nur geänderte Daten. Durch diese Techniken, wird der Server von den Clients nur minimal belastet. Das Ergebnis ist, dass die Leistungsfähigkeit des Gesamtsystems mit jedem zusätzlichen Client steigt.

Diese Tatsache steht im klaren Kontrast zu einem typischen Client/Server-System, bei dem jeder zusätzliche Client die Leistungsfähigkeit des Servers bremst.

Jedoch nicht nur in der Topologie unterscheidet sich ***dbGonzales*** von den üblichen Datenbanken. Im Gegensatz zu einer relationalen Datenbank müssen die einzelnen Datensätze einer Tabelle nicht die gleiche Struktur haben. Eine ***dbGonzales*** Tabelle verwaltet die Datensätze objektorientiert, somit können in einer ***dbGonzales*** Tabelle Datensätze von einer Basis-Struktur, sowie von einer dieser Basis-Struktur abgeleiteten Struktur abgelegt werden (entsprechend den Klassen einer objektorientierten Programmiersprache).

Der Zugriff auf die Objekte in der Datenbank erfolgt in der jeweiligen Programmiersprache über spezielle Interface-Objekte. Die Erstellung und Zuordnung dieser Objekte wird von Datenbank-Schnittstelle erledigt. Somit ist es denkbar einfach mit den Daten-Objekten zu arbeiten.

Die Datenbank

Die ***dbGonzales***-Datenbank besteht aus einer hochoptimierten Engine mit folgenden Eigenschaften:

- Jede Datenbank besteht aus einer oder mehreren Storage-Areas. Dabei können Storage-Areas auf unterschiedlichen Servern verteilt werden.
- Die Daten werden auf Seiten (Pages) gehalten. Jede Page ist 8Kb groß. Jede Storage-Area kann bis zu 4 Milliarden Pages adressieren. Somit ergibt sich eine maximale logische Größe pro Storage-Area von 32 Terra-Byte.
- Die Datensätze werden auf den Pages verteilt. Jeder Datensatz ist dabei variabel in der Größe, sowie in der Anzahl der Felder. Ein Datensatz kann aber nicht die Größe einer Page überschreiten.
- Zur Indizierung verwendet ***dbGonzales*** ein besonders effizientes Prefix-B-Tree Verfahren.
- Transaktionen arbeiten in ***dbGonzales*** grundsätzlich ohne Blockierungen für andere Transaktionen. Um dies zu gewährleisten arbeitet ***dbGonzales*** mit mehrfachen Transaktions-Generationen.
- Schreibzugriffe werden auf dem Client über eine Cache ausgeführt. Erst wenn eine Schreiboperation bestätigt wird, werden die Daten entgültig zum Server übertragen. Dadurch reduziert sich der Netzverkehr und die Zeit in der eine Schreib-Transaktion auf dem Server geöffnet ist.

Veith System GmbH . Laiming 3 . 83112 Frasdorf . Germany
Tel: 08052-2636 Email: info@veith-system.de

- Zur schnellen Übertragung der Daten verwendet ***dbGonzales*** eine Technik, die Warp-I/O heißt. Warp-I/O entlastet das Netz und den Server bei der Übertragung der Daten. Benchmarks zeigen, dass ***dbGonzales*** selbst bei I/O intensiven Abfragen die Daten über das Netz schneller liest, als andere Datenbanken von einem lokalen Massenspeicher.
- Ein intelligenter Cache-Algorithmus überträgt nur geänderte Daten.
- Abfragen werden entweder über eine SQL-ähnliche Abfragesprache oder direkt in der Programmiersprache erstellt. Durch das lokale Zwischenspeichern der Daten, eignet sich ***dbGonzales*** für Aufgaben, die mit einer klassischen Datenbank nicht durchführbar sind.

Veith System GmbH . Laiming 3 . 83112 Frasdorf . Germany
Tel: 08052-2636 Email: info@veith-system.de