

# Für den Webshop neckermann.de ist MySQL unternehmenskritisch

VON JESUS VILLAR-RODRIGUEZ\*

Wir betreiben für neckermann.de die Business-to-Consumer-Plattform, sprich den Webshop für den gesamten deutschsprachigen Raum inklusive Benelux-Länder. Die Bedeutung des Shops lässt sich daran ablesen, dass die Neckermann Versand AG Anfang des Jahres in neckermann.de GmbH umbenannt wurde. Der strategische Weg zur e-Driven-Company zeigt sich auch darin, dass unser Umsatzanteil von derzeit etwa 30 Prozent in den nächsten Jahren vervielfacht werden soll.

## GPL-Lizenz nicht gewollt

Unser Online-Shop ist klassisch nach dem Lamp-Stack mit Linux-Betriebssystem, Apache-Webserver, den Skriptsprachen Perl/PHP und jetzt auch mit MySQL aufgebaut. Allerdings können wir nicht als reine Open-Source-Anwender gelten. Da die fachlichen Anforderungen immer Priorität haben, verwenden wir wie bei unserer internen Suchmaschine durchaus Closed-Source-Produkte. Auch bei MySQL haben wir nicht die GPL-Version gewählt, sondern vor allem wegen des Supports die Enterprise-Variante mit dem Gold-Support. Da wir den Code nicht verändern, wäre im Prinzip auch eine GPL-Lizenz denkbar, ist aber momentan nicht gewollt.

Vor MySQL hatten wir eine Sybase-Datenbank im Einsatz und machten damit durchaus gute

Erfahrungen. Das System passte aber nicht mehr zu unserem neuen Architekturmodell. Sybase-Systeme laufen bevorzugt auf großen Servern und wachsen nach dem Scale-up-Konzept. Wir haben uns aber für Scale-out mit Intel-basierenden Blade-Servern entschieden. Unsere Aufgabe war es, dafür zu sorgen, dass die Kosten nicht in demselben Maße steigen wie unser Wachstum auf der geschäftlichen Seite. Die PC-Blades sind im Preis-Leistungs-Verhältnis deutlich günstiger als die Unix-Server. Die Nachteile von Scale-out-Architekturen beim Platz- und Stromverbrauch schlagen bislang

**„Trotz des eingesetzten Lamp-Stacks sind wir keine reinen Open-Source-Anwender.“**

kaum zu Buche, weil Platz momentan preiswert ist und wir beim Verbrauch und der Abwärme noch in akzeptablen Bereichen liegen. Dennoch wird Energieeffizienz sicher ein Thema in der Zukunft sein.

Vorher arbeiteten wir mit großen 24-CPU-Unix-Servern von Sun, die jetzt durch 150 Intel-Server ersetzt sind. Darunter befinden sich 110 Blades, zwei Unix-Systeme und einige Application-Server. Das MySQL-Cluster besteht aus zehn Servern: zwei mal

vier Systeme als Datenknoten für Deutschland beziehungsweise für die Auslandstöchter, zudem zwei Management-Knoten. Die normalen MySQL-Datenbanken (Produktdaten, User-Daten etc.) laufen auf weiteren 22 Servern plus den Installationen auf den Application-Servern.

## Einteilung in Gruppen

Die Datenbanken sind, und das ist der architektonische Unterschied zu unserem Konzept von vorher, vier logischen Gruppen zugeordnet, die wir nach Anforderungs- und Zugriffsmustern gebildet haben. Anders als bei den klassischen Datenbanken gehört es zum Konzept von MySQL, für verschiedene Aufgaben unterschiedliche Datenbank- und Speicher-Engines integrieren zu können.

■ Das Cluster (mit der NDB-Engine) verwenden wir für Session-Daten, also dort, wo hohe Schreib- und Lesezugriffe erforderlich sind. Hier kommt die Eigenschaft des MySQL-Clusters zum Tragen, dass die Datenbank (rund 700 MB) komplett im Hauptspeicher gehalten wird, was hohe Geschwindigkeiten garantiert.

■ Das normale MySQL versorgt den Shop, der ja vor allem Lesezugriffen ausgesetzt ist, mit etwa einem GB an Produktdaten. Hier kommt die für diese Anwendung optimierte Speicher-Engine MyISAM zum Einsatz.

■ Bei den rund 800 MB User-Daten brauchen wir Transaktionssicherheit und setzen daher auf InnoDB als Speicher-Engine.

■ Hinzu kommt ein Bereich für sonstige Daten im Umfang von 5 bis 6 GB, wo es weniger um kritische Anwendungen geht, sondern häufig um Projekte wie derzeit etwa Web 2.0. Hier arbeitet eine Kombination aus InnoDB und MyISAM.

Diese logische Trennung ermöglicht es uns, die Datenbankleistung genau dort zu optimieren, wo wir sie brauchen. Das funktionierte bei Sybase nicht. Dafür hatte Sybase andere Fea-

**„Das bislang verwendete Sybase-System eignete sich nicht für unseren Wechsel auf Scale-out.“**

tures, die erst für die kommenden Version 5.1 von MySQL angekündigt sind – insbesondere die Federated Storage Engine für verteilte Datenhaltung. Damit werden wir einen fünften Bereich einrichten, um über Reporting-Tools auf verschiedene Datentöpfe zugreifen zu können.

Wir sehen es sehr positiv, dass MySQL die Business-Eigenschaften der Datenbank ausbaut, auch wenn es nach unserem Geschmack schneller gehen könnte. Die Falcon-Transaktions-En-

gine mit den Acid-Eigenschaften, die erst irgendwann nach der Version 5.1 kommen soll, könnten wir schon heute brauchen, etwa für Data Warehousing, wo unser Unternehmen derzeit auf Oracle setzt. Allerdings weiß ich nicht, ob MySQL schon ausreichend Partner neben Business Objects dafür gewonnen hat.

## Oracle für Data Warehousing

Lücken gibt es aus unserer Sicht auch beim Backup. Wir verwenden den Legato Networker, für den aber kein Plug-in zu MySQL existiert, so dass wir Skripte schreiben lassen mussten. Momentan geht das, aber wenn ich an die Informationsmengen eines Data Warehouse denke, brauchen wir dafür vermutlich eine richtige Schnittstelle.

Eine Einbindung von MySQL in andere Unternehmensanwendungen wie ERP oder HR gibt es nicht. Die Anbindung an solche Backend-Systeme erfolgt über eine selbst programmierte Java-Middleware. Die eigentliche Verbindung läuft über die Applikation. Die maximale Größe eines Datensatzes ist bei MySQL Cluster eingeschränkt. Das führt bei Videos zu Problemen. Deshalb fahren wir Videos nicht über das Cluster. (ue) ♦

\*JESUS VILLAR-RODRIGUEZ

ist Teamleiter Websysteme/Backend Neue Medien bei der neckermann.de GmbH