



Frühe Ergebnisse bei Verbundoperationen

Diplomarbeit von Michael Cammert¹

Betreuer: Prof. Dr. Bernhard Seeger²

Zusammenfassung

In der Geschichte der Datenbanksysteme wurden zur Berechnung von Verbundoperationen (Joins) verschiedene Techniken entwickelt, darunter der einfache Nested-Loops-Join, der Sort-Merge-Join und der Hash-Join. U. a. mit zunehmender Bedeutung von Data-Warehouse-Systemen kam der Wunsch nach einer Anfragebearbeitung auf, die auch bei längeren Berechnungen früh Ergebnisse und Schätzungen über das Gesamtergebnis liefert. Da Joins bei Anfragen sehr häufig benötigt werden, mußten die Joinverfahren entsprechend angepaßt werden, was zunächst von Haas und Hellerstein für den Nested-Loops- und den Hash-Join geleistet wurde (*Ripple Joins for Online Aggregation*, SIGMOD, 1999) und diverse Weiterentwicklungen nach sich zog. Allerdings sind Algorithmen auf Basis von Nested-Loops nur bei komplett im Hauptspeicher ablaufenden Joins sinnvoll, während hashbasierte Joinverfahren nicht alle Arten von Joinprädikaten verarbeiten können.

Mit dem Progressive-Merge-Join (PMJ) wurde von Dittrich, Seeger, Taylor und Widmayer ein sortierbasierter Algorithmus entwickelt, der frühe Ergebnisse und Schätzer ermöglicht und diese somit dem erheblich größeren Spektrum der sortierbasierten Joinarten erschließt (*Progressive Merge Join: A Generic and Non-Blocking Sort-Based Join Algorithm*, VLDB, 2002). Damit eröffnete sich ein neues Feld für Erweiterungen und Optimierungen, das in dieser Diplomarbeit untersucht wird.

Beim Sort-Merge-Join werden zunächst die Eingaberelationen nacheinander extern sortiert, wozu jeweils Abschnitte in Hauptspeichergröße eingelesen, sortiert und wieder hinausgeschrieben werden. Der PMJ teilt dagegen den Hauptspeicher auf und sortiert dort je einen Abschnitt der Relationen, um dann vor dem Hinausschreiben deren Join zu berechnen. Dies liefert frühe Ergebnisse und ermöglicht eine Schätzung der Selektivität des Joins.

Als Weiterentwicklung des PMJ wird in dieser Arbeit der mehrdimensionale Progressive-Merge-Join (MPMJ) vorgestellt, der auch asymmetrische Joins von mehr als zwei Eingaberelationen berechnen kann. Dazu werden jeweils Abschnitte aller Relationen nebeneinander im Hauptspeicher sortiert und rekursiv deren Join berechnet.

Die Verteilung des Hauptspeichers unter den Relationen hat dabei wesentlichen Einfluß auf die Erzeugung früher Ergebnisse. Diese kann z.B. bezüglich Produktionsrate oder Gesamtzahl optimiert werden, welche noch erhöht werden kann, indem die sortierten Folgen nicht mehr gemeinsam, sondern nacheinander ausgetauscht werden, wobei zwischendurch jeweils der Join der im Hauptspeicher befindlichen Abschnitte berechnet wird.

Messungen bestätigen die Fähigkeit des MPMJ zur Produktion früher Ergebnisse und die Erhöhung ihrer Anzahl durch Einsatz der Optimierungen bei moderat erhöhter Gesamtlaufzeit.

¹cammert@informatik.uni-marburg.de

²seeger@informatik.uni-marburg.de