Kalibrierung von Kostenmodellen für föderierte DSMS

Michael Daum
md@cs.fau.de

Frank Lauterwald
frank.lauterwald@cs.fau.de

Philipp Baumgärtel
philipp.baumgaertel@cs.fau.de

Klaus Meyer-Wegener
kmw@cs.fau.de

Abstract: Bei verteilten Datenstromsystemen ist es ebenso wie bei verteilten Datenbanksystemen für die Verteilung von Anfragen entscheidend, die zu erwartenden Kosten schätzen zu können. Bei der Kostenschätzung mit Kostenmodellen müssen die Parameter für jedes System und jeden Operator ausgemessen werden. In dieser Arbeit wird ein *black-box*-Verfahren vorgestellt, mit dem es möglich ist, für beliebige Datenstromsysteme die Parameter des Kostenmodells auszumessen, und falls kein Kostenmodell vorliegt, nichtparametrische Modelle für Operatoren aufzustellen und auszumessen.

1 Einleitung

In verteilten Szenarien der Datenstromverarbeitung ist es notwendig, eine kostenoptimale Anfrageverteilung durchführen. Um eine kostenoptimale Anfrageverteilung durchführen zu können, wird ein Kostenschätzer benötigt. Im Bereich der Kostenmodellierung von Datenstromanfragen sind die ratenbasierten Modelle am weitesten verbreitet [VN02]. Weiterführende Arbeiten im Bereich der Kostenmodellierung von Datenstromsystemen sind [CKSV08], [GC06] und [LP06].

Kostenmodelle enthalten neben stromabhängigen Variablen, wie z.B. Selektivität und Raten, auch Konstanten, die von den einzelnen *Data Stream Management System*s (DSMSs) und der verwendeten Hardware abhängig sind. Um eine Kostenschätzung durchführen zu können, müssen nicht nur die Werte der Variablen bei der Ausführung einer Anfrage geschätzt werden, sondern zusätzlich müssen die systemspezifischen Konstanten bereits im Vorfeld bestimmt worden sein. In föderierten Umgebungen wie DSAM [DLF+10] ist es notwendig, mehrere teils unterschiedliche Datenstromsysteme auszumessen.

Ein möglicher Ansatz, die Konstanten der Kostenmodelle zu bestimmen, wäre es, sie aufgrund von Informationen bezüglich der Implementierung und dem Detailwissen über die verwendete Hardware abzuschätzen. Dies ist nicht praktikabel, da nicht für jedes System der Quellcode zur Verfügung steht. Außerdem ist eine genaue Modellierung der Hardwareeigenschaften und der Implementierungsdetails sehr aufwändig und fehleranfällig.

Einfacher lassen sich die Konstanten der Kostenformeln bestimmen, indem Messungen an realen Systemen durchgeführt werden. Diese Messreihen können dann benutzt werden, um mittels Ausgleichsrechnung eine Abschätzung der Konstanten zu erhalten. Falls für

einen Operator kein Kostenmodell zur Verfügung steht, so können die Messreihen mit Hilfe eines nichtparametrischen Ansatzes für eine Kostenschätzung verwendet werden.

Für heterogene und verteilte Datenbanksysteme wurden bereits mehrere Ansätze zur Abschätzung von Kosten entwickelt. Einige davon basieren auf der Kalibrierung von Kostenmodellen. Diese Kalibrierungsansätze lassen sich aufgrund der Unterschiede der Kostenmodelle nicht direkt auf Datenstromsysteme anwenden; das Prinzip der Modellerstellung und Kalibrierung mit Testanfragen ist jedoch ähnlich. In [Kos04] wird ein Überblick über verschiedene Ansätze zur Kostenschätzung bei heterogener und verteilter Anfrageverarbeitung gegeben. Dabei wird auch ein Kalibrierungsansatz vorgestellt.

In [DKS92] wird ein Verfahren vorgestellt, das durch Testanfragen und Messungen Rückschlüsse auf die Verarbeitung von Anfragen in Datenbanksystemen ohne Kenntnis der Implementierungsdetails erlaubt. Eine andere Methode zur Kalibrierung von Kostenmodellen ist die Klassifikation von Testanfragen [ZL94]. Klassifikationskriterien sind dabei Informationen über die verwendeten Tabellen, Fähigkeiten der zugrundeliegenden *Database Management System*s (DBMSs) und Eigenschaften der Anfrage, wie z.B. die Verwendung eines Joins. Für jede Klasse von Anfragen wird eine Kostenformel entworfen, die dadurch kalibriert wird, dass Testanfragen aus dieser Klasse ausgeführt werden und deren Kosten gemessen werden.

In [ZL96] wird detailliert beschrieben, wie für die Anfrageklassen aus [ZL94] Kostenmodelle erstellt und kalibriert werden können. Weiterhin wird beschrieben, wie die Kostenmodelle in einen Hauptteil zur Grobabschätzung der Kosten und einen Ergänzungsteil zur Berücksichtigung von Faktoren mit geringerem Einfluss zur genaueren Abschätzung aufgespalten werden können. Es wird beschrieben, wie auf einfache Weise bei Messungen Ausreißer entdeckt werden können, indem alle Werte, die weiter als die vierfache Standardabweichung vom Mittelwert für eine bestimmte Anfrageklasse entfernt sind, als Ausreißer angesehen werden. Als weiteres Problem der Kalibrierung wird Multikollinearität sowie Methoden zu ihrer Erkennung beschrieben.

Dieser Beitrag zeigt im Folgenden, wie mittels Kostenmessungen von Testanfragen an DSMSs Kostenmodelle kalibriert werden können. Zusätzlich wird ein Verfahren vorgestellt, das ohne Kostenmodelle auskommt. Beide Verfahren wurden mit Hilfe eines kommerziellen Datenstromsystems erfolgreich evaluiert.

2 Kalibrierung von Kostenmodellen

In diesem Abschnitt wird erläutert, auf welche Weise Modelle kalibriert werden können, wenn gemessene Datensätze vorhanden sind. Es wird von einem Operator ausgegangen, dessen Kostenmodell n metrische Variablen $x_1, \ldots, x_n \in \mathbb{R}$ enthält, die Eigenschaften des Stroms oder der Anfrage repräsentieren. Zusammen ergeben die Variablen den Vektor $\boldsymbol{x} = (x_1, \ldots, x_n)$. Durch Messungen wurden für m Variablenvektoren $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ die Kosten $y_1, \ldots, y_m$ des Operators gemessen.

2.1 Parameterbestimmung von Kostenmodellen

Es wird von einer Kostenfunktion $f(\boldsymbol{x}, \boldsymbol{\beta})$ ausgegangen, mit dem Variablenvektor $\boldsymbol{x}$ und einem Vektor $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$, der die zu bestimmenden Konstanten enthält. Die von der Kostenfunktion geschätzten Kosten seien $y = f(\boldsymbol{x}, \boldsymbol{\beta})$. Damit die einzelnen Konstanten bestimmt werden können, müssen mehr Messwerte als Konstanten vorliegen, es muss also $m > k$ gelten. In [ZL94] wird angegeben, dass $m \geq 10 \cdot (k + 1)$ sein sollte.

Gesucht ist das $\boldsymbol{\beta}$, welches die Summe der Fehlerquadrate $S = \sum_{i=1}^{m} (y_i - f(\boldsymbol{x}_i, \boldsymbol{\beta}))^2$ minimiert. Da nicht davon ausgegangen werden kann, dass die Konstanten nur linear in die Kostenformel f einfließen, kann keiner der linearen Lösungsansätze für die Anwendungen der „Methode der kleinsten Quadrate“ auf dieses Problem angewandt werden. Eine mögliche Lösung ist der in [Mar63] beschriebene Algorithmus, mit dem sich iterativ der Konstantenvektor $\boldsymbol{\beta}$ bestimmen lässt, der S minimiert.

Die Anwendung der „Methode der kleinsten Quadrate“ ist problematisch, wenn in den Messdaten lineare Abhängigkeiten vorhanden sind. Dies ist der Fall, wenn einzelne Variablen in den Variablenvektoren $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ linear oder näherungsweise linear von anderen Variablen abhängig sind. Diese sogenannte Multikollinearität wird in [Sil69] beschrieben. In [FG67] wird beschrieben, wie sich Multikollinearität detektieren und lokalisieren lässt.

2.2 Kalibrierung von nichtparametrischen Modellen

Unter Umständen kann es vorkommen, dass für einen Operator kein Kostenmodell entwickelt wurde oder dass sich ein System nicht entsprechend verhält. Statt eine neue Modell zu entwickeln bzw. ein bestehendes anzupassen, ist ein generisches Modell, das jedem Systemverhalten angepasst werden kann, eine Alternative. Durch Interpolation der gemessenen Kosten eines Operators für bestimmte Parameter können die Kosten für andere Belegungen der Parameter geschätzt werden. Da die zu den Messwerten gehörenden Parameter eventuell nicht gleichförmig verteilt sind, bietet sich die Interpolation mit *Radial Basis Function Network*s (RBFNs) aus [PG89] an.

Problematisch ist dabei einerseits, dass unter Umständen sehr viele Messwerte vorliegen. Dies führt zu einem hohen Rechenaufwand bei der Interpolation und zu einem großen Modell, da alle Messwerte benötigt werden, um das Modell darzustellen. Andererseits sind die Messwerte fehlerbehaftet und somit ist eine exakte Interpolation nicht sinnvoll.

Diese Probleme lassen sich dadurch lösen, dass der in [PG89][Abschnitt 4.4] beschriebene Ansatz zur Approximation von Funktionen durch RBFN verwendet wird. Dafür werden ℓ Punkte $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_\ell \in \mathbb{R}^n$ gewählt mit $\ell < m$. Diese Punkte sollten aus dem Bereich des $\mathbb{R}^n$ gewählt werden, in dem auch die Messwerte liegen. Die einfachste Methode, diese Punkte zu bestimmen, ist laut [PG89], zufällig ℓ Messwerte zu wählen. Um die Kostenfunktion darzustellen, werden also nur die Punkte $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_\ell$ und die zugehörigen Koeffizienten $c_1, \ldots, c_\ell$ benötigt. Da bei dieser Methode keine interpolierende Kurve durch die Punkte sondern es wird eine glatte Kurve durch die möglicherweise fehlerbehafteten Messwerte gelegt wird, kann ein Algorithmus zum Auffinden von Ausreißern sinnvoll sein.

2.3 Auffinden von Ausreißern in den Messwerten

Ausreißer in den Messdaten sind sowohl bei der Parameterbestimmung von Kostenmodellen als auch bei der Kalibrierung von nichtparametrischen Modellen problematisch. Diese Ausreißer vor der Kalibrierung eines Modells zu bestimmen ist nicht möglich, da ohne Modellkurve nicht bestimmt werden kann, wie weit ein Messwert von dem erwarteten Wert entfernt ist. Dieses Problem lässt sich durch die Anwendung des *Random Sample Consensus* (RANSAC) Algorithmus aus [FB81] lösen.

2.4 Bestimmung der Kosten einzelner Operatoren

Da die meisten Systeme, für die ein Kostenmodell kalibriert werden soll, nur selten detaillierte Informationen über die Kosten einzelner Operatoren zur Verfügung stellen, ist ein Vorgehen nötig, mit dem diese Kosten von außen bestimmt werden können. Bei Performance-Messungen kann nur das Verhalten des Systems als Ganzes gemessen werden. In diesem Abschnitt wird erläutert, wie eine Messreihe entworfen werden muss, damit auf die Kosten eines einzelnen Operators geschlossen werden kann.

Es wird angenommen, dass für die Kalibrierung eines Operators bereits eine Messreihe mit m Messungen entworfen wurde. Das bedeutet, dass Testdatensätze $d_1, \ldots, d_m$ und Anfragen $q_1, \ldots, q_m$ generiert wurden. Diese Daten und Anfragen wurden so gewählt, dass für die Tests $i \in \{1, \ldots, m\}$ die Variablenvektoren $\boldsymbol{x}_i$ des Operators bestimmte Werte annehmen, so dass mit den Kosten des Operators $y_1, \ldots, y_m$ ein Modell kalibriert werden kann. Wie sich diese Kosten des Operators aus den gemessenen Gesamtkosten bestimmen lassen, soll im Folgenden erläutert werden.

Ebenso wird vorausgesetzt, dass die Kosten eines Operators unabhängig von der Anzahl und Auslastung anderer Operatoren im System sind. Das bedeutet, dass sich die Kosten der einzelnen Operatoren und die Grundkosten des Systems zu den Gesamtkosten addieren lassen. Dies ist die Grundvoraussetzung für Kostenmodelle, die nur die Kosten einzelner Operatoren schätzen, ohne andere Operatoren zu berücksichtigen, die das System bearbeitet.

Um nun für Test i die Kosten des einzelnen Operators zu bestimmen, muss dieser Test mehrfach mit der gleichen Anzahl an Ein- und Ausgabeadaptern und unterschiedlicher Anzahl an Instanzen des Operators mit den gleichen Parametern durchgeführt werden. Ist o die Anzahl der Operatorinstanzen, sind γ_i die Grundkosten des Systems inklusive Ein- und Ausgabeadapter und sind y_i die Kosten einer Operatorinstanz bei Test i, so gilt für die Gesamtkosten $c_i(o) = \gamma_i + y_i \cdot o$.

Die Gesamtkosten $c_i(o)$ sind von außen messbar. Anschließend können mit der Methode der kleinsten Quadrate sowohl γ_i als auch die gesuchten Einzelkosten des Operators y_i geschätzt werden. Dabei muss allerdings beachtet werden, dass γ_i nicht die wirklichen Grundkosten des Systems sind, sondern auch die Kosten der Ein- und Ausgabeadapter enthalten.

Ein weiterer Punkt, der beachtet werden muss, ist das Verhalten des Kostenwerts, wenn sich dieser der maximalen Systemkapazität nähert. Beispielsweise ist anzunehmen, dass sich die CPU-Last nur bis zu einem bestimmten Grenzwert linear in Abhängigkeit der Operatoranzahl verhält.

Zu erläutern ist noch, wie der Operatorgraph strukturiert sein muss, so dass mehrere Instanzen des gleichen Operators mit denselben Parametern darin vorkommen ohne dass die Anzahl der Ein- und Ausgabeadapter verändert werden muss. Für die Anordnung von o Operatoren bieten sich zwei grundlegende Ansätze an. Einerseits können die Operatoren in Reihe und andererseits können sie parallel angeordnet sein. Die Anordnung der Operatoren in Reihe kommt der tatsächlichen Struktur im Operatorgraph am nächsten, da in diesem Fall der Ausgabestrom eines Operators wieder der Eingabestrom des nächsten Operators ist. Diese Struktur ist allerdings zum Testen nicht geeignet, da Operatoren die Eigenschaften des Stroms ändern und es so nicht immer möglich ist, dass jeder Operator in der Reihe mit den gleichen Testparametern ausgeführt wird.

Problematisch ist die Latenz. Diese ist nur dann linear von der Anzahl der Operatoren abhängig, wenn diese in Reihe geschalten sind. Wenn die Latenz eines einzelnen Operators bestimmt werden soll, müsste ein Weg gefunden werden, diesen Operator in einer Weise mehrfach hintereinander auszuführen, so dass sich die Latenzen addieren. Für parallel angeordnete Operatoren ist im Idealfall eine konstante Latenz zu erwarten.

Die Grundkosten des Systems lassen sich bestimmen, indem von den gemessenen Gesamtkosten eines Operatorgraphen die Kosten der einzelnen Operatoren und Adapter abgezogen werden. Dies sollte für mehrere Anfragegraphen durchgeführt werden. Auf diese Weise lässt sich die Güte der Kostenschätzung evaluieren.

3 Evaluation

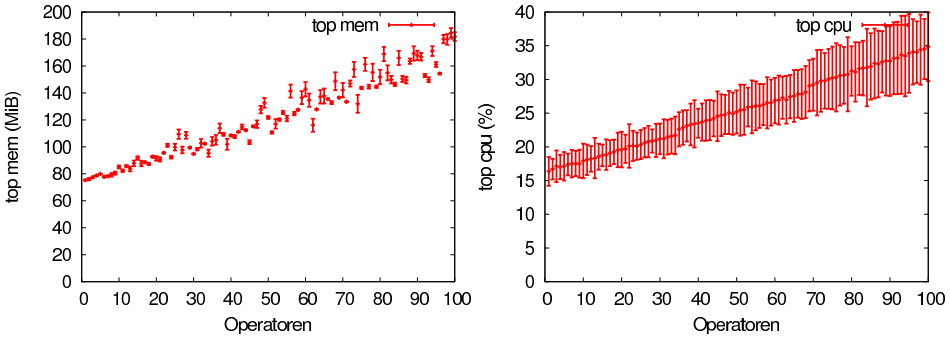
Zur Evaluation der Kalibrierung von Kostenmodellen wurde ein kommerzielles Datenstromsystem verwendet. Für jede Parameterbelegung wurde eine entsprechende Anfrage erzeugt, das Datenstromsystem wurde gestartet und die Anfrage wurde 2 Minuten lang unter konstanten Bedingungen ausgeführt. Es stellte sich heraus, dass die Messwerte der ersten Minute teilweise unbrauchbar waren, weshalb stets nur die Messwerte für die zweite Minute der Messungen aggregiert werden. Ein einzelner veröffentlichter „Messwert“ für eine Parameterbelegung ist also der Durchschnitt der Messwerte der zweiten Minute des Testlaufs. Die Standardabweichung der Messwerte wird in den Abbildungen als Fehlerbalken dargestellt.

3.1 Operatorkosten

Zur Bestimmung einzelner Operatorkosten wurde zunächst untersucht, wie sich die Gesamtkosten in Abhängigkeit von der Anzahl der Operatoren verhalten. Die Rate betrug bei diesem Test 5000 Tupel pro Sekunde. Die Operatoren werden dabei in einer Anfrage

parallelgeschaltet. Alle Operatoren verwenden als Eingabe den gleichen Strom desselben Inputreaders. Nur der Ausgabestrom des ersten Operators wird von einem Outputwriter ausgegeben. Die anderen Ausgabeströme werden verworfen.

Die Abbildung 1 zeigt das erwartete lineare Verhalten, so dass die Kosten eines einzelnen Operators der Steigung der Geraden, die durch die Messwerte gelegt werden kann, entsprechen.

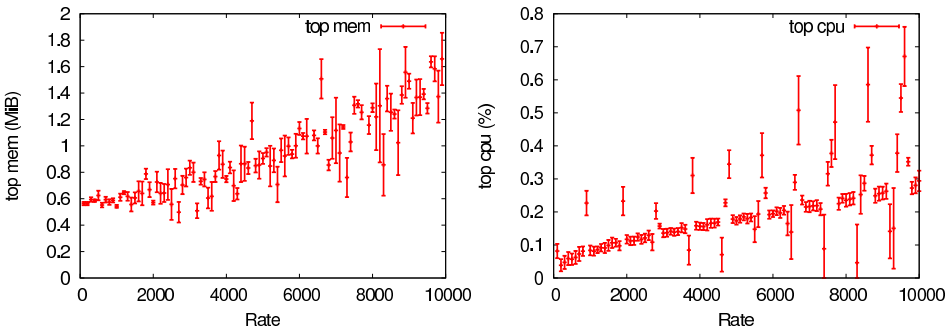


(a) Speicherverbrauch (b) CPU–Last

Abbildung 1: Speicherverbrauch und CPU–Last des Filters mit `top` gemessen

Bei der Latenz zeigte sich nicht das im Idealfall zu erwartende konstante Verhalten, sondern ein nichtlineares Ansteigen in Abhängigkeit von der Anzahl der Operatoren. Dies ist eine Folge des Scheduling. Aus diesem Grund ist für einen Operatorgraphen in Reihenschaltung auch keine lineare Abhängigkeit der Latenz von der Anzahl der Operatoren zu erwarten. Die Latenz eines einzelnen Operators lässt sich also nicht mit den in dieser Arbeit beschriebenen Methoden bestimmen. Dazu müsste eine genauere Betrachtung mittels der Warteschlangentheorie durchgeführt werden.

In Abbildung 2 werden die Einzelkosten des Filteroperators für den gemessenen Speicherverbrauch und die CPU–Last dargestellt. An Stellen, an denen die Gesamtkosten des



(a) Speicherverbrauch (b) CPU–Last

Abbildung 2: Einzelkosten des Filters (Speicherverbrauch und CPU–Last mit `top` gemessen)

Systems „Sprünge“ aufweisen, entstehen bei der Bestimmung der Einzelkosten Ausreißer,

die noch entfernt werden können (Abschnitt 2.3).

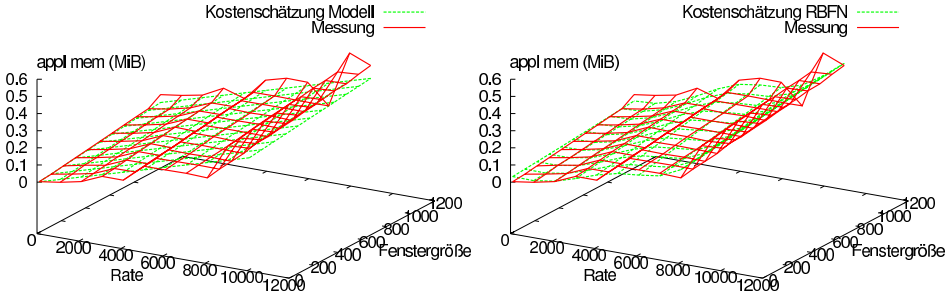
3.2 Modellkalibrierung

Das verwendete Datenstromsystem stellt einen Statusstrom zur Verfügung, der zum Monitoring des Systems verwendet werden kann. Für die Kalibrierung von Kostenmodellen wird exemplarisch der *Application Memory* aus diesem Statusstrom für den Aggregatoperator betrachtet. Der Speicherverbrauch war in diesem Fall präziser zu ermitteln als mit einer externen Messung. Der Speicherverbrauch des Aggregatoperators wird aufgeteilt in den Speicher, der für die Warteschlange des Operators benötigt wird, und in den Speicherbedarf des Operators. Der Speicherbedarf der Warteschlange ist das Produkt aus der mittleren Warteschlangenlänge und der Größe eines Tupels M_{Tupel}. Laut dem Gesetz von Little ist die mittlere Warteschlangenlänge gleich dem Produkt aus der Rate λ und der mittleren Wartezeit $W^{(1)}$. Der Speicherbedarf des Fensters des Aggregatoperators ist das Produkt aus Fenstergröße s und der Größe eines Tupels M_{Tupel}.

$$M_{\text{Gesamt}} = \underbrace{\lambda \cdot M_{\text{Tupel}} \cdot W^{(1)}}_{=:M_0} + s \cdot M_{\text{Tupel}} \tag{1}$$

Die Anpassung der Gleichung 1 an die Messwerte des Tests ergibt $M_0 \approx 37,4$ KiB und $M_{\text{Tupel}} \approx 92,7$ KiB.

In Abbildung 3(a) sind die Messwerte zusammen mit den durch das kalibrierte Kostenmodell geschätzten Werten eingezeichnet. Dabei ist zu erkennen, dass das Kostenmodell die tatsächlichen Kosten sehr gut abschätzt. Die Anwendung der Methode der Funktionsapproximation aus Abschnitt 2.2 zeigt in Abbildung 3(b), dass die Funktionsapproximation eine glatte Kurve durch die Messpunkte erzeugt. Die Sprünge in den Messwerten werden besser approximiert, als durch das einfache Kostenmodell.

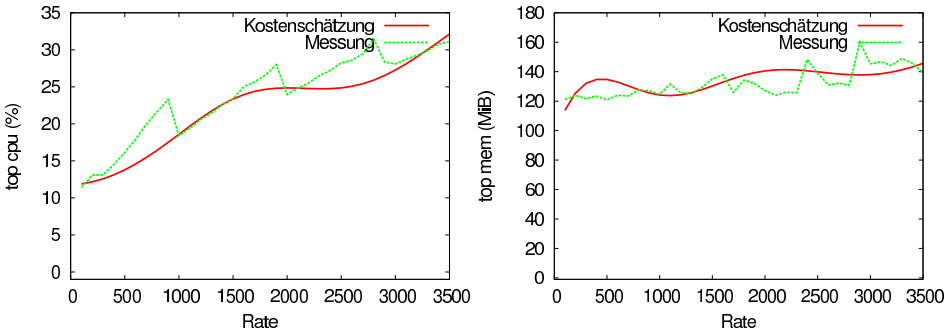


(a) Kostenmodell (b) Funktionsapproximation

Abbildung 3: Kalibriertes Kostenmodell und Funktionsapproximation des Aggregatoperators

3.3 Kostenschätzung am Beispiel

Es bleibt zu überprüfen, ob die Kosten mit Modellen geschätzt werden können. Dafür wurden für die Kosten des Inputreaders, Outputwriters und Filters mittels RBFN Funktionsapproximationen durchgeführt. Die Funktionsapproximation wurde gewählt, um zu testen, inwieweit Sprünge in den gemessenen Kosten abgeschätzt werden können. Die Grundkosten des Systems wurden bestimmt, indem von den gemessenen Kosten eines Operatorgraphs die Kosten der einzelnen Operatoren und Adapter abgezogen wurden.



(a) CPU–Last (b) Speicherverbrauch

Abbildung 4: Schätzung der CPU–Last und des Speicherverbrauchs von 100 Filteroperatoren mittels RBFN

Für einen Operatorgraph, bestehend aus 100 Filtern, wurde eine Kostenschätzung durchgeführt. Anschließend wurden die tatsächlichen Kosten testweise gemessen. In Abbildung 4 werden die geschätzten Kosten mit den gemessenen Kosten verglichen. Es lässt sich erkennen, dass die Kosten gut abgeschätzt werden können. Allerdings zeigt die Kostenschätzung mittels RBFN teilweise Schwankungen. Dies liegt an der Wahl der Stützstellen und der dem RBFN zugrundeliegenden Funktion. In weiteren Arbeiten müsste untersucht werden, wie sich die Funktionsapproximation weiter optimieren ließe.

4 Bewertung und Ausblick

Mit Hilfe von Testanfragen und Testdaten können Messreihen erstellt werden , die der Kalibrierung von Kostenmodellen dienen. Es wurde erklärt, wie ausgehend von den Messdaten für ein DSMS die Kosten eines einzelnen Operators bestimmt werden und wie mit diesen Werten ein Kostenmodell kalibriert werden kann. Außerdem wurde ein nichtparametrisches Modell entwickelt, das für die Kostenschätzung verwendet werden kann. Es kann angewendet werden, wenn für einen Operator eines Systems entweder keine Kostenformel bekannt ist oder wenn sich dessen Kosten anders verhalten, als dies durch das Kostenmodell vorhergesagt wurde. Diese Ansätze wurden mit Hilfe eines kommerziellen Datenstromsystems evaluiert.

Datenstromsysteme werden dabei als Black Box gesehen, über deren interne Funktions-

weise nur wenige Annahmen getroffen werden. Neben den Operatorkosten können außerdem die Grundkosten des Systems bestimmt werden. Die Bestimmung der Kosten ganzer Operatorgraphen ist problemlos möglich, falls die Werte der operatorspezifischen Variablen bekannt sind. Wie diese z.B. für Raten ermittelt werden können, wurde bereits in [DLBMW10] beschrieben. Um praktisch einsetzbar zu sein, müsste das hier beschriebene Verfahren allerdings noch systematisch für alle Kostenwerte und Operatoren verschiedener Datenstromsysteme durchgeführt werden.

Ein weiteres Problem der Kostenschätzung bei heterogenen und verteilten Systemen ist die Abhängigkeit der Kosten von der Hardware. Dies führt dazu, dass ein Kostenmodell für jede verwendete Hardwarekonfiguration neu kalibriert werden müsste, was hohen Aufwand verursacht. Interessant wäre eine Möglichkeit, Kostenmodelle so zu kalibrieren, dass diese Modelle für unterschiedliche Hardware verwendet werden können. In [KK06] wird die Idee vorgestellt, die Kosten eines Operators mit einem Hardwarespezifischen Performance–Index zu multiplizieren.

Dies würde auch für nichtparametrische Modelle funktionieren, da auch hier die approximierte Funktion mit dem Performance Index skaliert werden kann. Unter Umständen ist eine einfache Skalierung mit einem Faktor allerdings zu grobgranular, um die Eigenschaften der Hardware zu repräsentieren. Eine Erweiterung des Performance–Index wäre eine feingranulare Berücksichtigung der Hardwareeigenschaften bereits im Kostenmodell. Das Kostenmodell müsste dann für einige Hardwarekonfigurationen kalibriert werden, um für unterschiedliche Hardware allgemeingültig zu sein.

Analog zu der Verwendung von Hardwareparametern in Kostenmodellen könnte auch eine nichtparametrische Schätzung in Abhängigkeit von Hardwareeigenschaften durchgeführt werden. Dabei müsste die Erstellung eines nichtparametrischen Modells für unterschiedliche Hardwarespezifikationen durchgeführt werden. Anschließend können die Kosten für Systeme, die auf einer anderen Hardware ausgeführt werden, durch Interpolation bestimmt werden.

Eine Alternative zu nichtparametrischen Modellen wäre symbolische Regression [Koz92]. Dabei wird ausgehend von Messdaten mittels genetischer Algorithmen eine Formel erstellt, die möglichst gut die von Messdaten beschriebene Kurve beschreibt. Auf diese Weise könnte für ein DSMS automatisiert eine Kostenformel erstellt werden, die sich einfacher für unterschiedliche Hardware kalibrieren ließe als die nichtparametrischen Modelle.

Literatur

[CKSV08] Michael Cammert, Jurgen Kramer, Bernhard Seeger und Sonny Vaupel. A Cost-Based Approach to Adaptive Resource Management in Data Stream Systems. *Transactions on Knowledge and Data Engineering*, 20(2):230–245, Feb. 2008.

[DKS92] Weimin Du, Ravi Krishnamurthy und Ming-Chien Shan. Query Optimization in a Heterogeneous DBMS. In *VLDB '92: Proceedings of the 18th International Conference on Very Large Data Bases*, Seiten 277–291, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[DLBMW10] Michael Daum, Frank Lauterwald, Philipp Baumgärtel und Klaus Meyer-Wegener. Propagation of Densities of Streaming Data within Query Graphs. In *Proceedings of the 22nd International Conference on Scientific and Statistical Database Management (SSDBM)*, Seiten 584–601, 2010.

[DLF+10] Michael Daum, Frank Lauterwald, Martin Fischer, Mario Kiefer und Klaus Meyer-Wegener. *Integration of Heterogeneous Sensor Nodes by Data Stream Management*, Kapitel Wireless Sensor Network Technologies for Information Explosion Era, Seiten 139–172. Number 278 in Studies in Computational Intelligence. Springer, 2010.

[FB81] Martin A. Fischler und Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[FG67] Donald E. Farrar und Robert R. Glauber. Multicollinearity in Regression Analysis: The Problem Revisited. *The Review of Economics and Statistics*, 49(1):92–107, 1967.

[GC06] J. Gomes und H.A. Choi. Cost-based Solution for Optimizing Multi-join Queries over Distributed Streaming Sensor Data. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006*, 2006.

[KK06] Richard Kuntschke und Alfons Kemper. Data Stream Sharing. In *Current Trends in Database Technology - EDBT 2006*, 2006.

[Kos04] D. Kossmann. The State of the Art in Distributed Query Processing. *ACM Computing Surveys (CSUR)*, 32(4):422–469, 2004.

[Koz92] John R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA, USA, 1992.

[LP06] Y. Liu und B. Plale. Multi-Model Based Optimization for Stream Query Processing. In *KSI Eighteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'06)*, 2006.

[Mar63] Donald W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[PG89] Tomaso Poggio und Federico Girosi. A Theory of Networks for Approximation and Learning. Techreport, Massachusetts Institute of Technology, Cambridge, MA, USA, 1989.

[Sil69] S. D. Silvey. Multicollinearity and Imprecise Estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 31(3):539–552, 1969.

[VN02] Stratis Viglas und Jeffrey F. Naughton. Rate-Based Query Optimization for Streaming Information Sources. In *ACM SIGMOD Conference (SIGMOD)*, Seiten 37–48, 2002.

[ZL94] Qiang Zhu und Per-Åke Larson. A Query Sampling Method of Estimating Local Cost Parameters in a Multidatabase System. In *Proceedings of the Tenth International Conference on Data Engineering*, Seiten 144–153, Washington, DC, USA, 1994. IEEE Computer Society.

[ZL96] Qiang Zhu und Per-Åke Larson. Building regression cost models for multidatabase systems. In *DIS '96: Proceedings of the fourth international conference on on Parallel and distributed information systems*, Seiten 220–231, Washington, DC, USA, 1996. IEEE Computer Society.