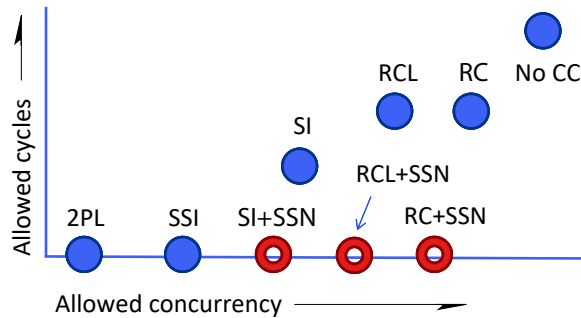


# The Serial Safety Net: Efficient Concurrency Control on Modern Hardware

Seminarausarbeitung

Florian Lüdiger

Technische Universität Dortmund  
florian.luediger@tu-dortmund.de



**Abbildung 1: Klassische Concurrency-Control-Verfahren im Vergleich zum Serial Safety Net**

## 1 EINLEITUNG

In diesem Dokument wird das in der Veröffentlichung „The Serial Safety Net: Efficient Concurrency Control on Modern Hardware“ von Wang et al. [2] vorgestellte Verfahren zur Sicherstellung der Serialisierbarkeit von Transaktionsplänen erläutert. Dabei werden bestehende Concurrency-Control-Verfahren wie beispielsweise Snapshot-Isolation(SI), Read-Committed(RC) oder Read-Committed mit Locks(RCL) so erweitert, dass diese die Serialisierbarkeit der entstehenden Pläne gewährleisten.

Der Vorteil des Serial Safety Nets(SSN) gegenüber klassischen Verfahren, wie dem Zwei-Phasen-Sperrprotokoll(2PL) oder der Serializable-Snapshot-Isolation(SSI), besteht darin, dass eine bessere Nebenläufigkeit von Transaktionen ermöglicht wird, wodurch der Durchsatz des gesamten Systems massiv gesteigert wird.

In Abbildung 1 wird schematisch dargestellt, dass ein bestimmter Trade-off zwischen der zugelassenen Nebenläufigkeit und den erlaubten Zyklen im Abhängigkeitsgraphen, also dem gewünschten Isolationslevel, besteht. Klar erkennbar ist, dass ein optimales Concurrency-Control-Verfahren keinerlei Zyklen erlaubt und dennoch eine maximale Nebenläufigkeit gewährleistet. Es wird deutlich, dass die durch das Serial Safety Net erweiterten Verfahren überhaupt keine solcher Zyklen erlauben und somit in dieser Hinsicht gleichwertig zu Verfahren wie 2PL und SSI sind. Gleichzeitig ist allerdings erkennbar, dass die erlaubte Nebenläufigkeit wesentlich höher ist und somit ein Performanzgewinn erwartet wird.

Einige Gründe dafür, dass die bisher bekannten Verfahren zur Sicherstellung der Serialisierbarkeit, wenig performant sind, finden sich darin, dass diese einen hohen Overhead produzieren und teilweise zu unnötigen Transaktionsabbrüchen führen, wodurch möglicherweise gültige Pläne verworfen werden. Außerdem funktionieren diese aufgrund schlechter Skalierung schlecht auf moderner Hardware, bei der immer mehr Operationen im Hauptspeicher stattfinden, wodurch die I/O-Last sinkt und damit ein noch größerer Fokus auf einem effizienten Concurrency-Control-System liegt.

Wichtig ist in diesem Zusammenhang zu erwähnen, dass ein Transaktionsplan, der in diesem Dokument als serialisierbar bezeichnet wird, keinen Schutz gegen Phantomereignisse bietet. Das Serial Safety Net lässt sich durch das Verwenden von Sperren allerdings leicht erweitern, sodass das Vorkommen von Phantomereignissen ausgeschlossen wird, worauf allerdings hier nicht weiter eingegangen werden soll. Weitere Informationen zu möglichen Anomalien, Isolationsleveln und dem Phantom-Problem finden sich in [1].

## 2 THE SERIAL SAFETY NET

Das Serial Safety Net baut auf bestehenden Multiversion-Concurrency-Control-Verfahren auf. In einem System, welches Multiversion-Concurrency-Control(MVCC) verwendet, bestehen alle Datenbankelemente aus einer Sequenz von Versionen, wobei Schreiboperationen jeweils eine Version anlegen und Leseoperationen eine Version zurückgeben.

Für solche MVCC-Verfahren sichert das Serial Safety Net einen kreisfreien Abhängigkeitsgraphen, wodurch die Serialisierbarkeit des Transaktionsplans gewährleistet wird. Der Abhängigkeitsgraph stellt dabei eine Übersicht über die Abhängigkeit zwischen den Transaktionen eines Plans dar, wobei die Knoten des Graphen die committeten Transaktionen und die Kanten die Abhängigkeiten zwischen diesen darstellen. Ein Graph ohne Zyklen garantiert dabei immer, dass ein äquivalenter serieller Plan zu den ausgeführten Transaktionen existiert, welcher zum selben Ergebnis geführt hätte.

**DEFINITION 1.** Die Abhängigkeit  $T \leftarrow U$  zwischen den Transaktionen  $T$  und  $U$  besagt, dass  $U$  von  $T$  abhängig ist und somit  $T$  als direkter Vorgänger und  $U$  als direkter Nachfolger bezeichnet wird.

Eine zentrale Rolle bei der Umsetzung des Serial Safety Nets spielt die Untersuchung der Abhängigkeiten von Transaktionen in Verbindung mit der Commit-Reihenfolge. Dazu lassen sich zwei verschiedene Typen von Abhängigkeiten wie folgt definieren.

**DEFINITION 2.** Bei einer *Back-Edge*  $T \xleftarrow{b} U$  committet der Nachfolger  $U$  zuerst.

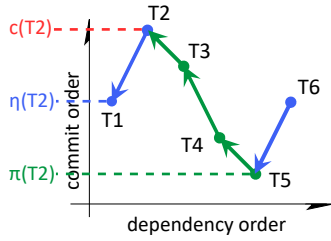


Abbildung 2: Veranschaulichung von **Back-** und **Forward-Edges** sowie den Zeitstempeln von Transaktion  $T_2$

DEFINITION 3. Bei einer **Forward-Edge**  $T \xrightarrow{f} U$  committet der Vorgänger  $T$  zuerst.

DEFINITION 4. Eine reflexive, transitive **Back-Edge**  $T \xleftarrow{b^*} U$  bezeichnet eine Verbindung, bei der  $T$  von  $U$  ausschließlich über Back-Edges erreichbar ist.

Zum besseren Verständnis sind die genannten Begriffe in Abbildung 2 veranschaulicht worden. Die abgebildeten Back-Edges ergeben zusammen eine reflexive, transitive Back-Edge von Transaktion  $T_2$  zu  $T_5$ .

Die Umsetzung des Serial Safety Nets erfordert das Erfassen der folgenden drei Zeitstempel zu jeder Transaktion, welche es später erlauben einen Abhängigkeitszyklus zu erkennen. Die vorgestellten Zeitstempel sind ebenfalls in Abbildung 2 für die Transaktion  $T_2$  eingezeichnet.

DEFINITION 5.  $c(T)$  bezeichnet die Commit-Zeit der Transaktion  $T$

DEFINITION 6.  $\pi(T)$  bezeichnet die Commit-Zeit des ältesten Nachfolgers, der durch Back-Edges erreichbar ist:

$$\pi(T) = \min(\{c(U) : T \xleftarrow{b^*} U\}) = \min(\{\pi(U) : T \xleftarrow{b} U\} \cup \{c(T)\})$$

DEFINITION 7.  $\eta(T)$  bezeichnet die Commit-Zeit des zuletzt committeten Vorgängers von  $T$ :

$$\eta(T) = \max(\{c(U) : U \xrightarrow{f} T\} \cup \{-\infty\})$$

## LITERATUR

- [1] Hal Berenson, Phil Bernstein, Jim Gray, Jim Melton, Elizabeth O'Neil, and Patrick O'Neil. 1995. *A Critique of ANSI SQL Isolation Levels*. Technical Report. <https://www.microsoft.com/en-us/research/publication/a-critique-of-ansi-sql-isolation-levels/>
- [2] Tianzheng Wang, Ryan Johnson, Alan Fekete, and Ippokratis Pandis. 2015. The Serial Safety Net: Efficient Concurrency Control on Modern Hardware. In *Proceedings of the 11th International Workshop on Data Management on New Hardware (DaMoN'15)*. ACM, New York, NY, USA, Article 8, 8 pages. <https://doi.org/10.1145/2771937.2771949>