

## 2. Übungsblatt

Ausgabe: 26. Oktober 2015 · Besprechung: 02. November 2015

### 1 Buffer Manager — Verdrängungsstrategien

In der Vorlesung wurde die Funktionsweise des *Buffer Managers* diskutiert. Dabei ist bei der Wahl eines zu überschreibenden *Frames* eine *Verdrängungsstrategie* notwendig.

Die wohl bekannteste Verdrängungsstrategie ist LRU (“Least Recently Used”). In Datenbanksystemen werden jedoch oft auch andere Strategien verwendet, etwa

- “*clock*” und
- “*LRU-k*” (*LRU-Variante, wobei der k-letzte Zugriff berücksichtigt wird*).

Wie funktionieren diese beiden Verdrängungsstrategien? Welche Vor- und Nachteile haben sie jeweils gegenüber einer reinen LRU-Strategie?

### 2 Buffer Manager und Strategien in der Praxis

Auf der Vorlesungswebseite finden Sie den Sourcecode für eine modifizierte Version von PostgreSQL 9.3.1. Dabei wurde der *Buffer Manager* so verändert, dass er `pin ()`- und `unpin ()`-Aufrufe jeweils protokolliert.<sup>1</sup>

1. Installieren Sie die so modifizierte PostgreSQL-Version auf Ihrem System. Aktivieren Sie das Protokollieren von Meldungen ab dem `DEBUG1`-Level, um die genannten Aufrufe sichtbar zu machen.
2. Modifizieren Sie Ihr Programm aus Aufgabe 1 so, dass es die `pin ()`- und `unpin ()`-Sequenz von PostgreSQL interpretieren und simulieren kann.
3. Lassen Sie Anfragen des TPC-H-Benchmarks im resultierenden System laufen. Untersuchen Sie die Effektivität (Hitrate) Ihrer gewählten Verdrängungsstrategie.

---

<sup>1</sup>Tatsächlich finden Sie nur die modifizierte Datei `bufmgr.c`. Diese wurde auf dem Webserver als `bufmgr.txt` abgelegt (weil das Content Management System des ITMC leider keine `.c`-Dateien erlaubt). Tauschen Sie diese bitte einfach in den Original-Quellen aus.