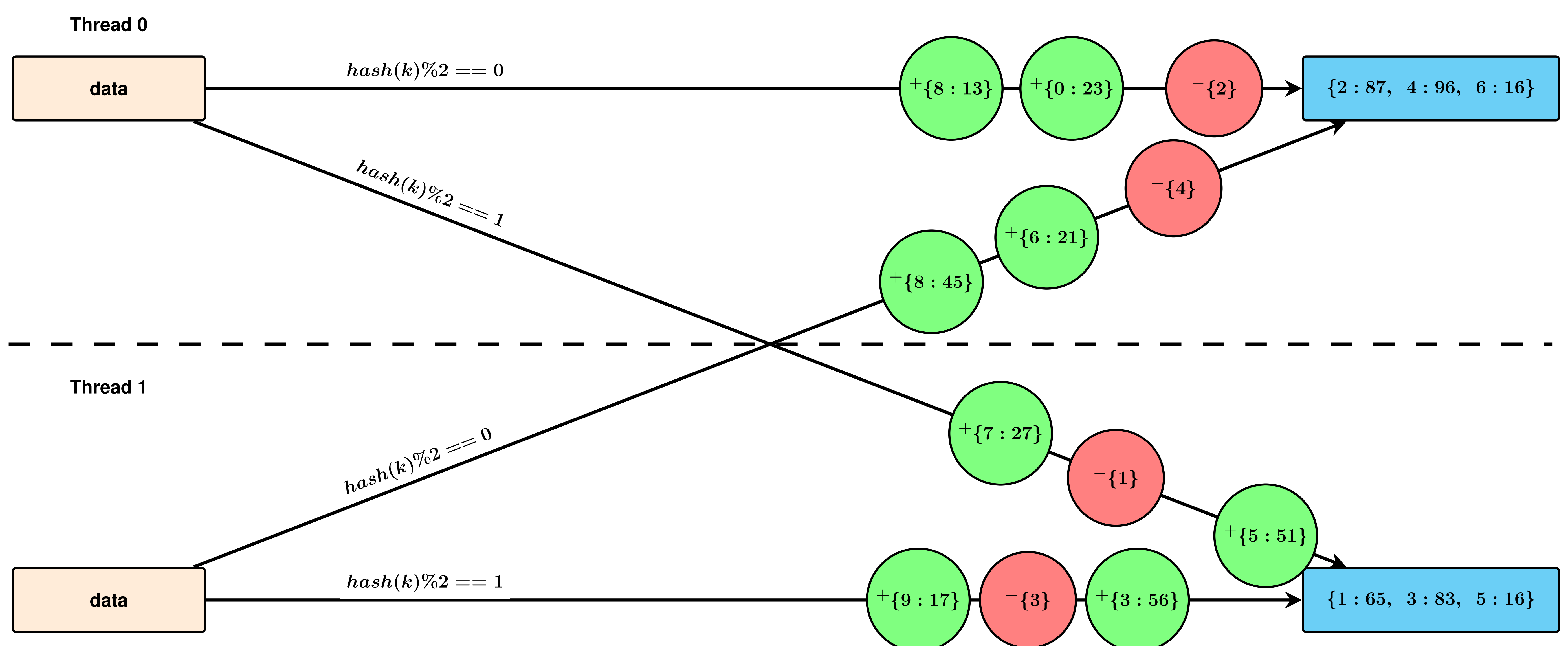


Lock-Free Concurrent Python/Numba Dictionary Mutation

Implementierung des ersten nebenläufigen Dictionary-Schreibzugriffs in Numba-kompatiblem Python-Code für parallel-beschleunigte Abschnitte



1. Problem

Schreibzugriffe auf Python-Dictionaries sind ein Flaschenhals.

Schreib- und Löschoptionen auf Dictionaries in Python benötigen oft den weit überwiegenden Teil der Rechenzeit, aber sie sind mit dem *Numba*-Parallel-Accelerator nicht parallelisierbar, falls das selbe Dictionary Zugriffsziel ist.

Numba bietet keine Mutex-Mechanismen für Code, der auf der CPU ausgeführt wird.

2. Lösungsansatz

Zusatz-Bibliothek entwickeln

1. Mehrere Dictionaries

Anstatt eines Dictionaries, werden so viele verwendet, wie parallel Threads abgearbeitet werden können. Die Daten werden nach einer festen Regel auf diese aufgeteilt und die Zugriffe erfolgen gleichzeitig.

2. Warteschlangen

Anstatt eines Mutex-Mechanismus verhindern threadübergreifende Warteschlangen den gleichzeitigen Schreibzugriff auf dasselbe Dictionary.

Funktionsweise

- Zuordnung des **Dictionaries** nach Hashwert des Keys
- Eintrag (Key und Value) + Befehl (**Add** oder **Delete**) wird in jeweilige Warteschlange eingeordnet
- Ein Schreib-Thread pro Dictionary entnimmt die Einträge aus zugehörigen Warteschlangen

Threads	Laufzeit (s)	CPU-Zeit (s)	Variante
1	1,80	1,80	konv.
1	3,70	3,70	par.
2	5,75	10,50	
7	1,60	11,20	
13	0,88	11,44	
30	0,60	18,00	

3. Keine Modifikation an Numba oder Python

Die Lösung soll möglichst unabhängig von vorhandenen Komponenten sein. Insbesondere sollen keine Änderungen an *Numba* erfolgen, damit sich der Wartungsaufwand in Grenzen hält.

3. Ergebnis

Bis 200% mehr Geschwindigkeit

Die gegenwärtige Implementierung reduziert die Zeit eines Benchmark-Laufes um ca. 33%, was einer Steigerung des Datendurchsatzes um 200% entspricht. Leider fällt die Effizienz aktuell auf 10% im Vergleich zur konventionellen Single-Thread-Implementierung

Testsystem

- Scaleway EM-I210E-NVME
- AMD EPYC 7313P (mit 1,5 GHz)

4. Nutzen

Mit *Numba* kompilierte Python-Software, welche eine erhebliche Zeit mit Schreibzugriffen auf Dictionaries verbringt, kann signifikant beschleunigt werden.

