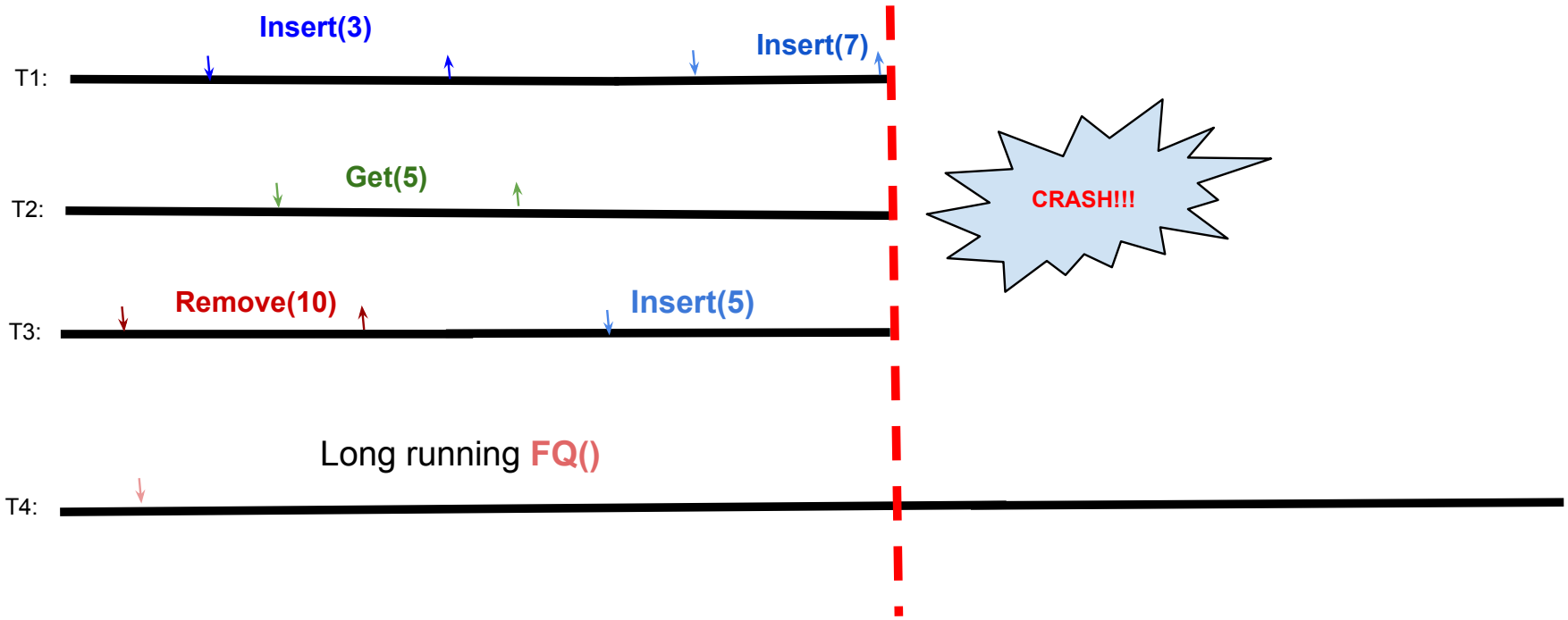
Efficient Resumable Filter Queries

Muktikanta Sa
Micron Technologies

Pierre Sutra
Telecom SudParis

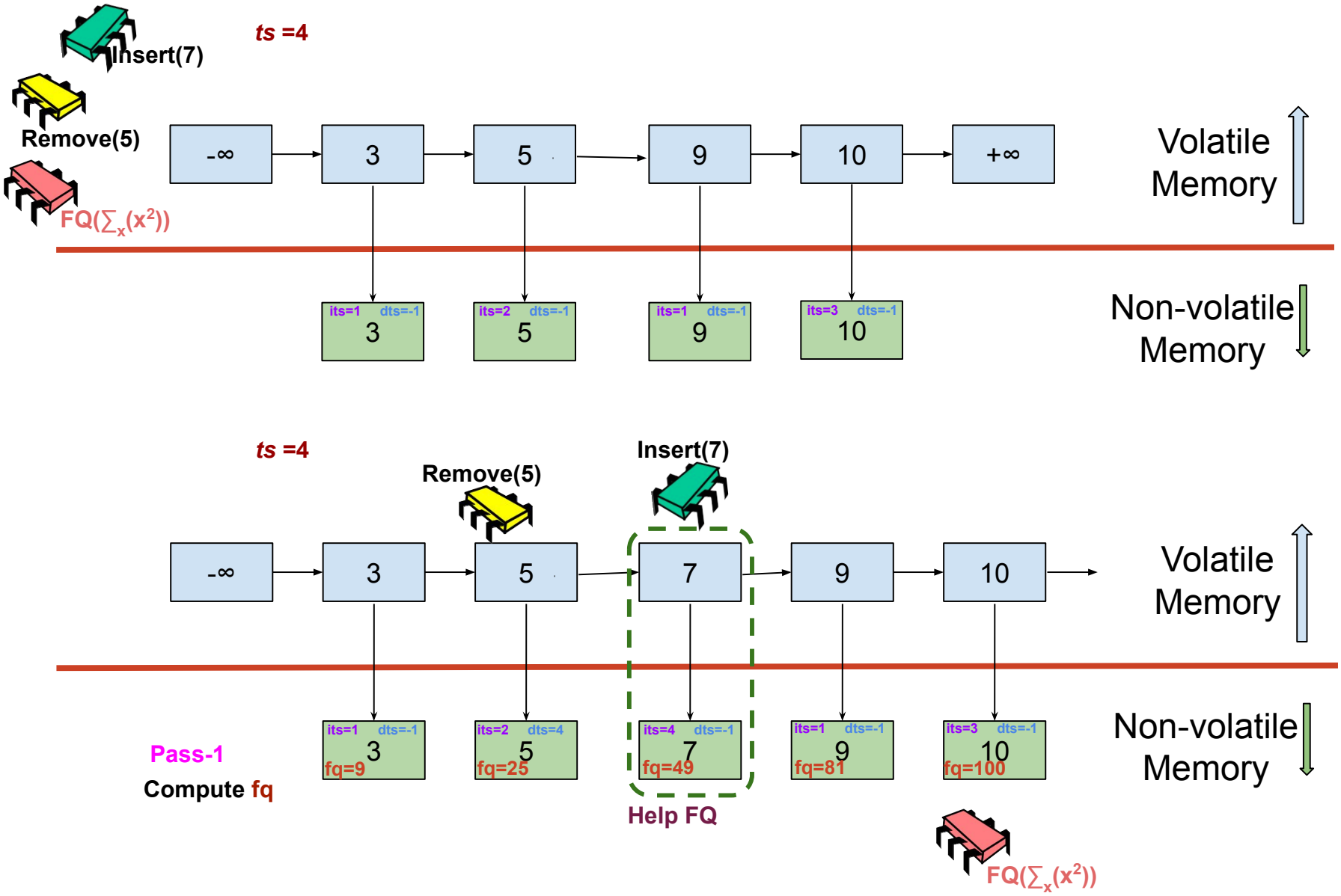
13th Annual Non-volatile Memories Workshop
May 2022

Motivation

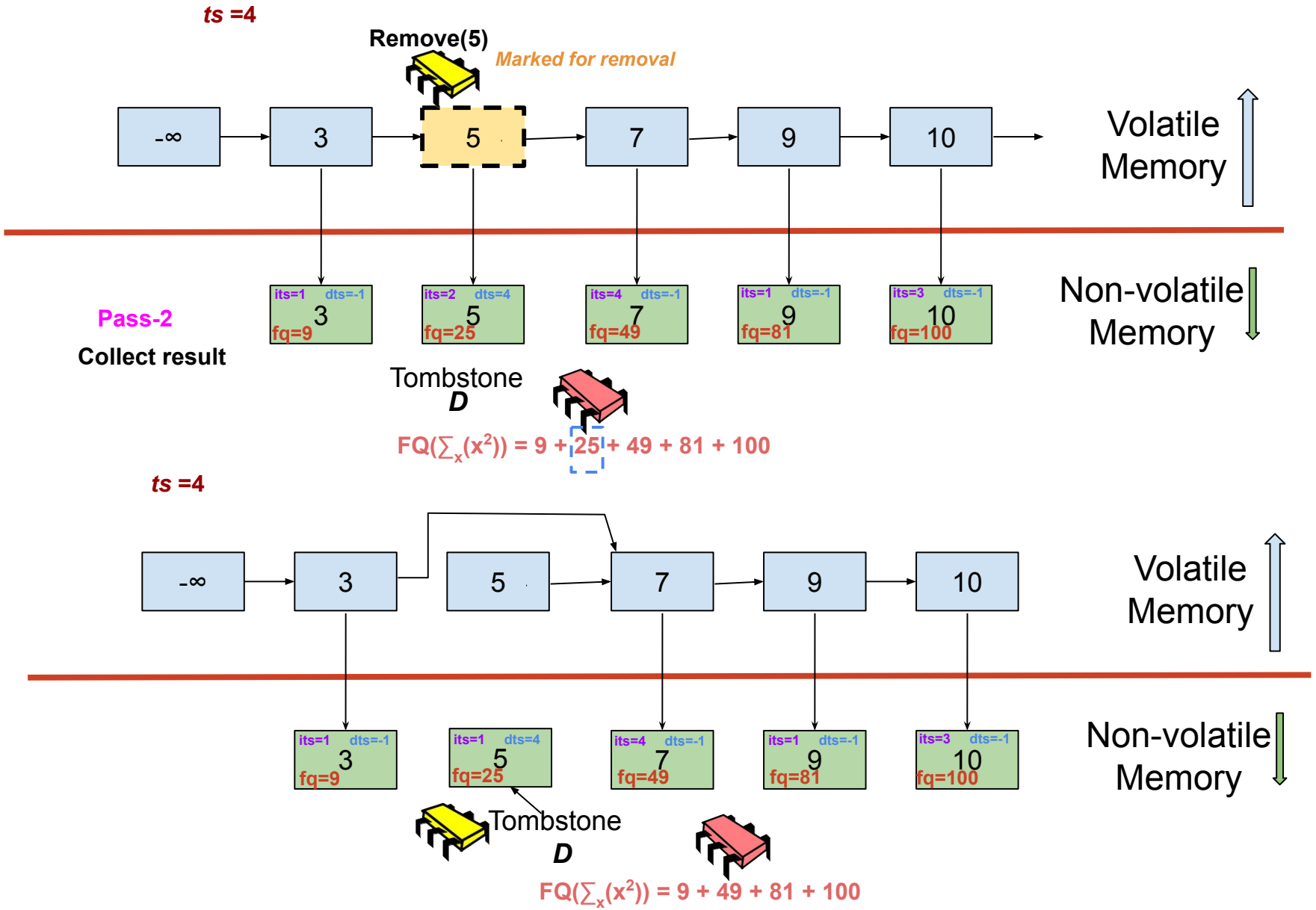


Problematic: How to resume a long-running task after a crash?

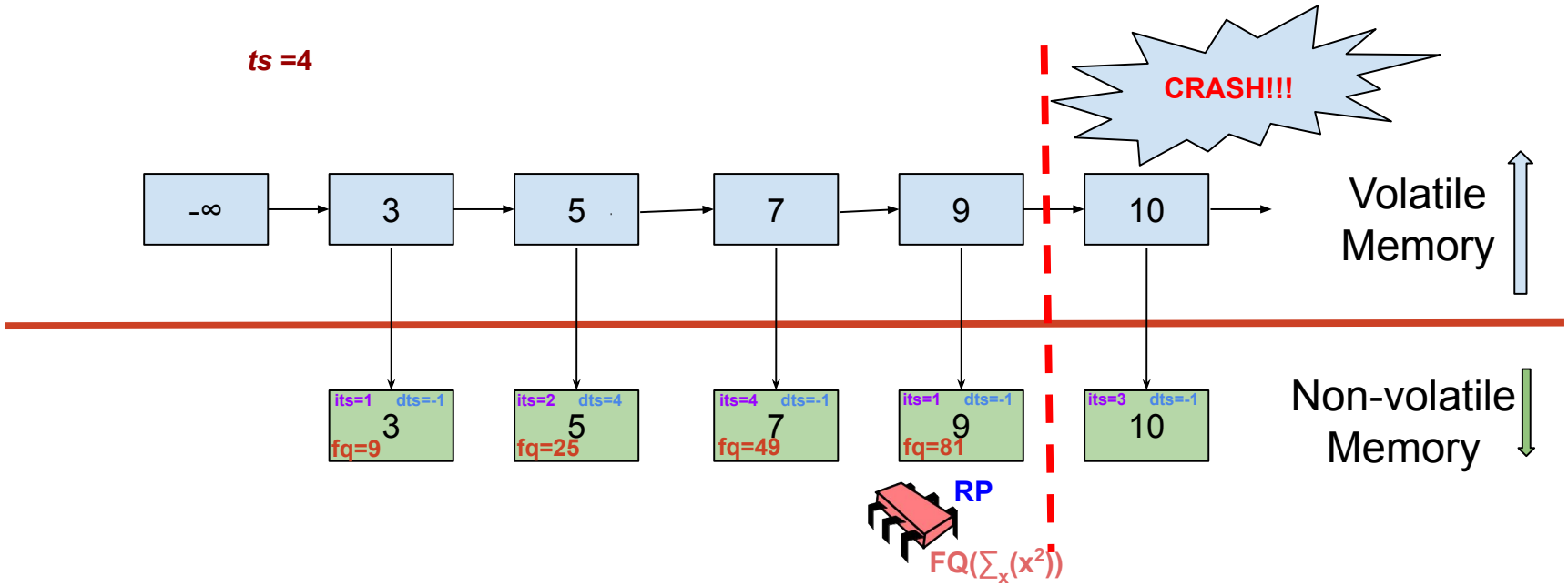
Key ideas

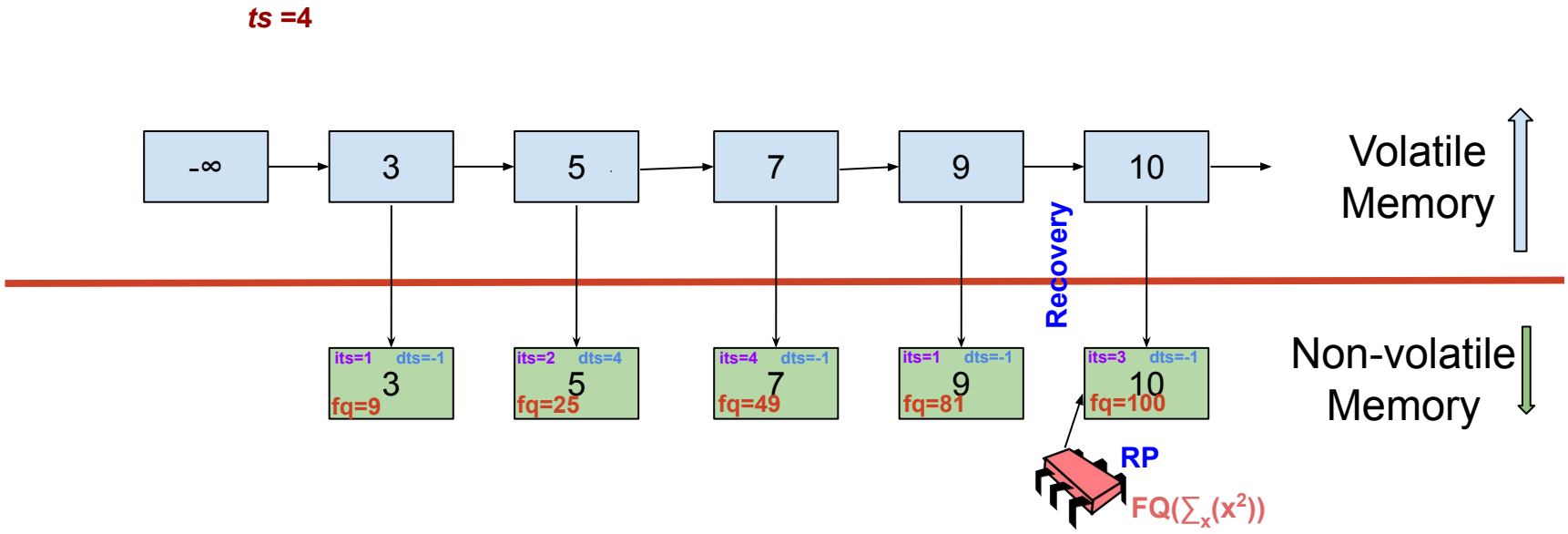

Insert(7)
ts =4
Remove(5)
FQ(∑x(x²))
-∞
3
5
9
10
+∞
Volatile Memory
Non-volatile Memory
its=1 dts=-1 3
its=2 dts=-1 5
its=1 dts=-1 9
its=3 dts=-1 10
ts =4
Remove(5)
Insert(7)
-∞
3
5
7
9
10
Volatile Memory
Non-volatile Memory
its=1 dts=-1 3 fq=9
its=2 dts=4 5 fq=25
its=4 dts=-1 7 fq=49
its=1 dts=-1 9 fq=81
its=3 dts=-1 10 fq=100
Pass-1
Compute fq
Help FQ
FQ(∑x(x²))

Key ideas

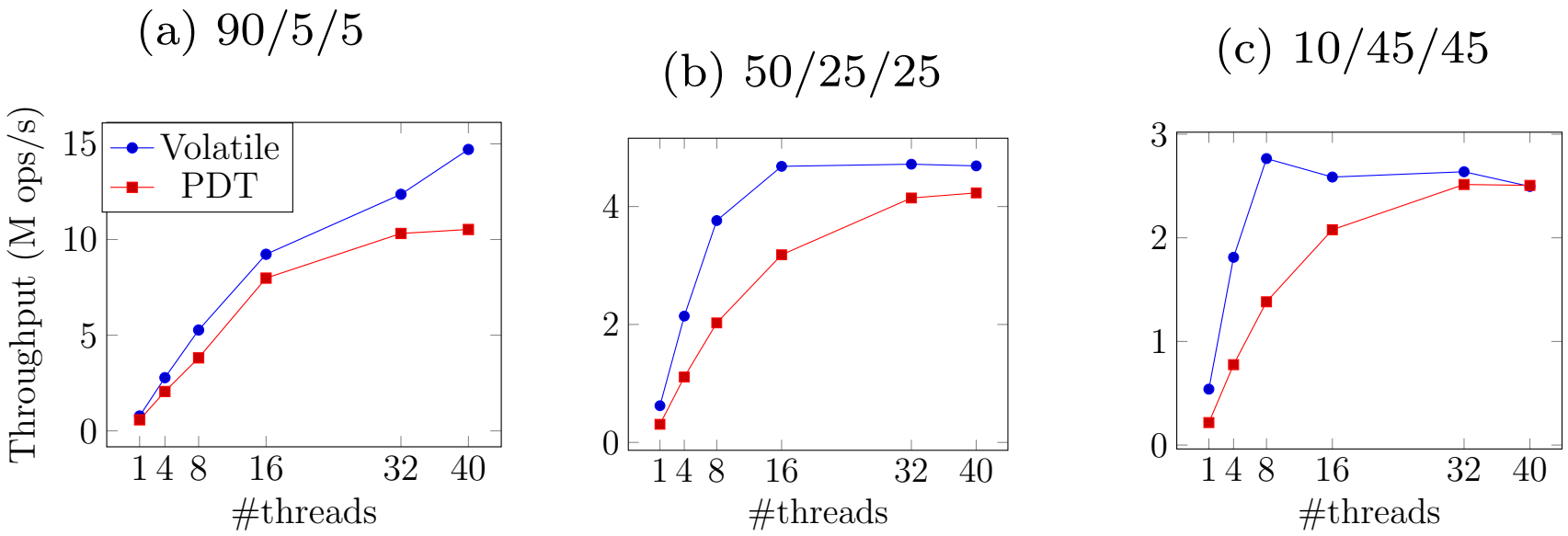

ts =4
Remove(5)
Marked for removal
-∞
3
5
7
9
10
Volatile Memory
Non-volatile Memory
its=1 dts=-1 3 fq=9
its=2 dts=4 5 fq=25
its=4 dts=-1 7 fq=49
its=1 dts=-1 9 fq=81
its=3 dts=-1 10 fq=100
Pass-2
Collect result
Tombstone
D
FQ(∑x(x2)) = 9 + 25 + 49 + 81 + 100
ts =4
-∞
3
5
7
9
10
Volatile Memory
Non-volatile Memory
its=1 dts=-1 3 fq=9
its=1 dts=4 5 fq=25
its=4 dts=-1 7 fq=49
its=1 dts=-1 9 fq=81
its=3 dts=-1 10 fq=100
Tombstone
D
FQ(∑x(x2)) = 9 + 49 + 81 + 100

Key ideas


ts =4
CRASH!!!
-∞
3
5
7
9
10
Volatile Memory
Non-volatile Memory
its=1 dts=-1 3 fq=9
its=2 dts=4 5 fq=25
its=4 dts=-1 7 fq=49
its=1 dts=-1 9 fq=81
its=3 dts=-1 10
RP
FQ(∑x(x2))


ts =4
-∞
3
5
7
9
10
Volatile Memory
Recovery
Non-volatile Memory
its=1 dts=-1 3 fq=9
its=2 dts=4 5 fq=25
its=4 dts=-1 7 fq=49
its=1 dts=-1 9 fq=81
its=3 dts=-1 10 fq=100
RP
FQ(∑x(x2))

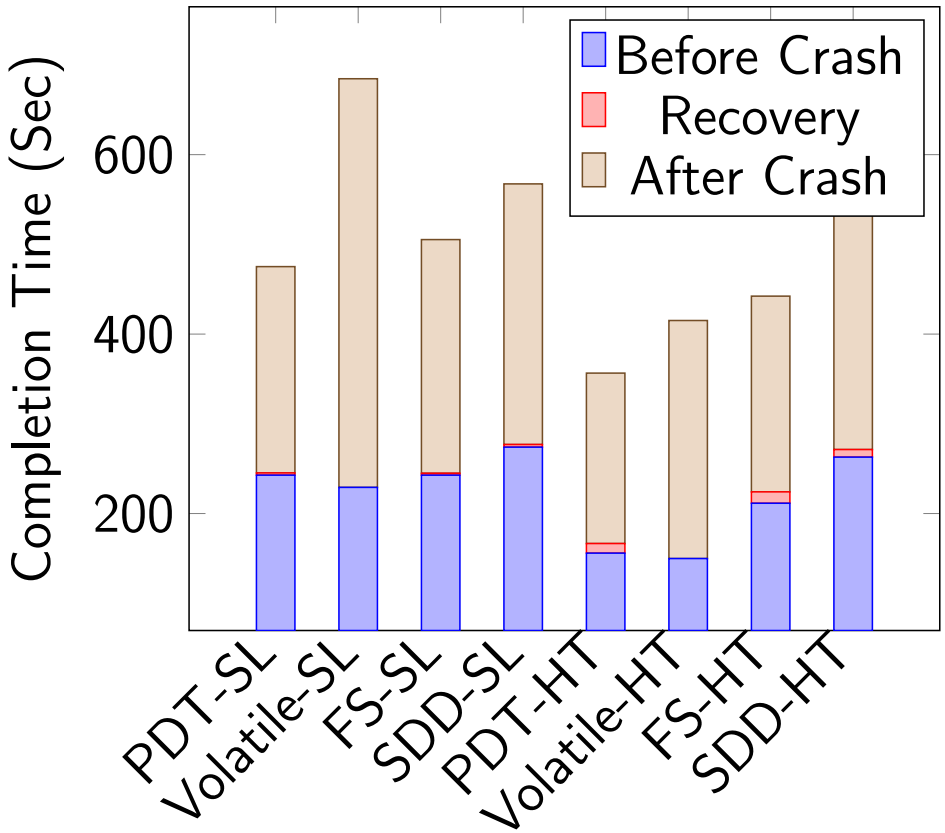
CRUD performance: Skiplist



Takeaways: Persistent skip list is slower than its volatile counterpart. Across all workloads, it is on average 5-10% slower.

Resumable Filter Query

GET:80%, INSERT:20%, and FQ is run by single thread.



Takeaways: **PDT-SL** is 40% faster than **Volatile-SL** and **PDT-HT** is 16% faster than **Volatile-HT**. Both also faster than a file system solution.

Conclusion and Future Works

- Durable Linearizability *inadequate* for long-running tasks
- Notion of *resumable* operations + *resumption points*
- Application to various PDTs (list, skip list and hash table)
  - w. support for resumable filter query
- Evaluation shows
  - performance close to volatile (5-25% slower)
  - save up to 40% computation time after a crash
- What's next:
  - use in datastores
  - test w. real workloads (e.g., YCSB-E, OLAP)