IBM SOLIDDB

In-Memory Database Optimized for Extreme Speed and Availability

Authors: Jan Lindstrom, Vilho Raatikka, Jarmo Ruuth, Petri Soini, Katriina Vakkila

Course Instructor: Stan Zdonik
Presenter: Lixiang (Gavin) Zhang

Outline:

Introduction

Speed - Vtrie

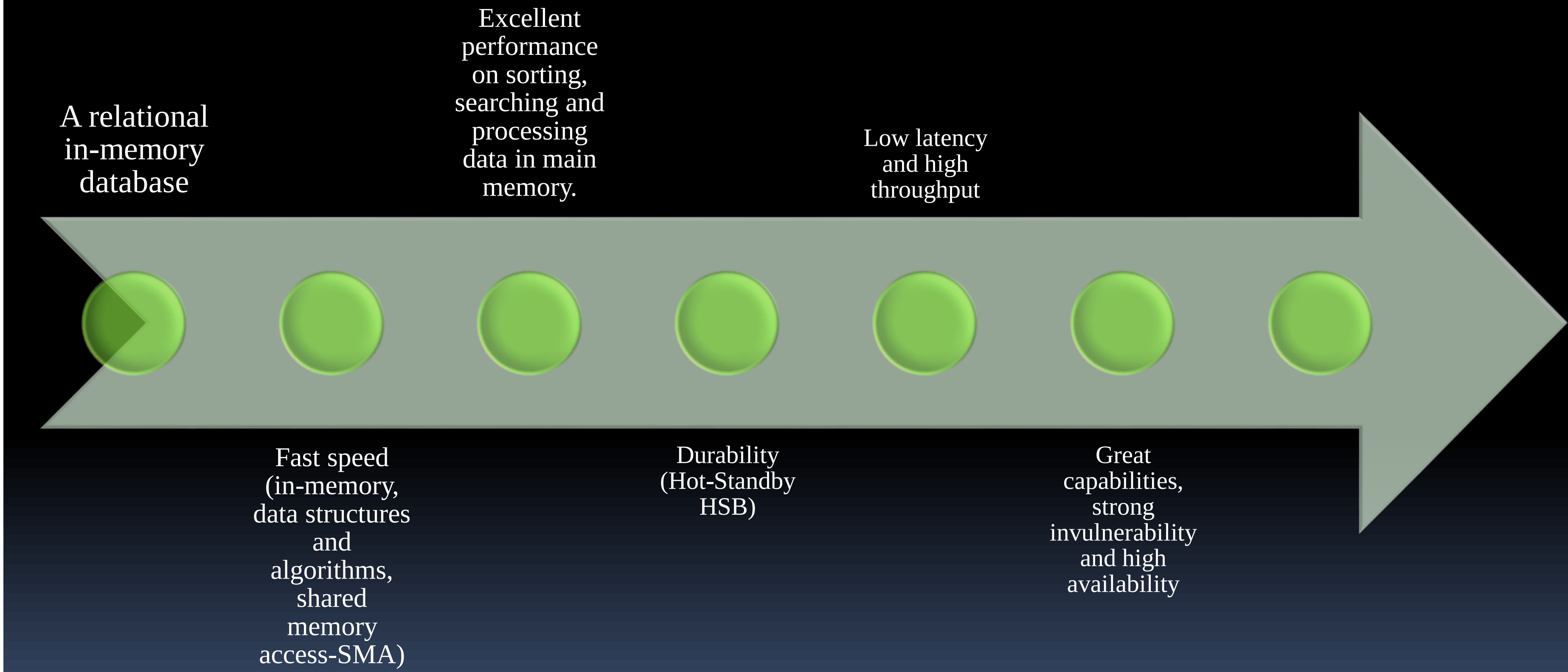
Recovery – Checkpoint

Availability – HSB

Concurrency control

Conclusion

Introduction:



A relational in-memory database

Fast speed (in-memory, data structures and algorithms, shared memory access-SMA)

Excellent performance on sorting, searching and processing data in main memory.

Durability (Hot-Standby HSB)

Low latency and high throughput

Great capabilities, strong invulnerability and high availability

Speed

Vtrie: Variable length trie (retrieval)

A trie is a multi-way tree structure that is widely used for storing strings.

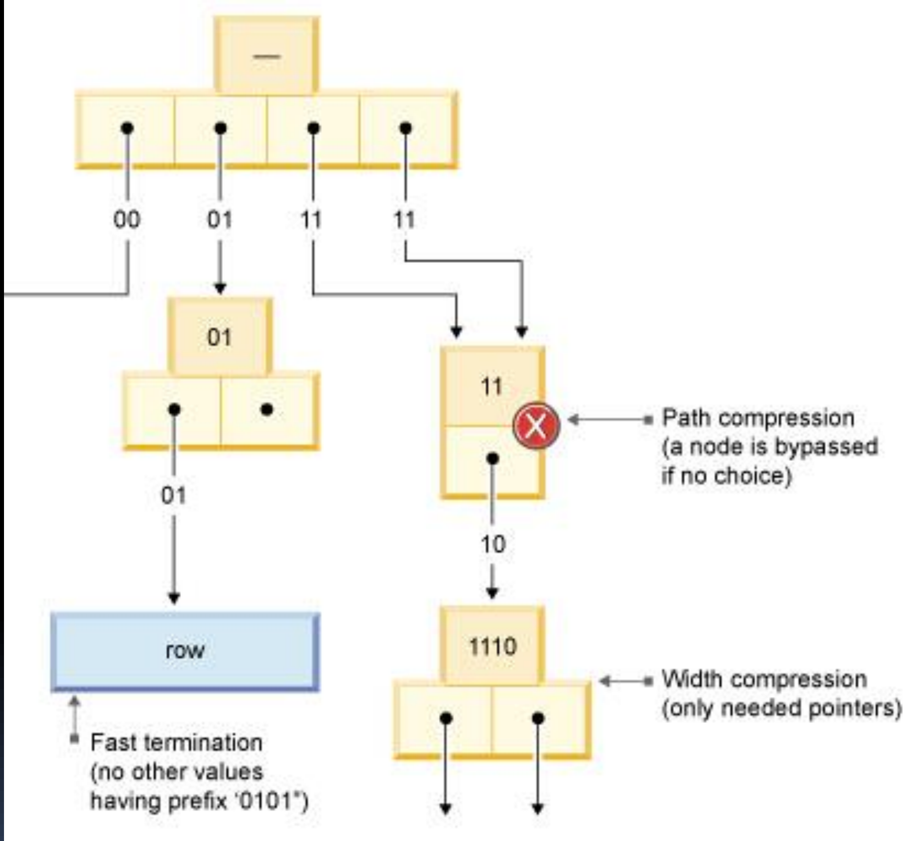
The idea is that all strings that share a common prefix hang off a common node.

VTrie uses bitwise tree where individual bits compose a key allowing keys to be any supported data type.

Vtrie does not execute any comparisons during tree traversal.

Each part of a key is applied as an array index to a pointer array of a child node.

Example of a VTrie structure (simplified)


—
00
01
11
11
01
11
01
10
row
1110
Path compression (a node is bypassed if no choice)
Width compression (only needed pointers)
Fast termination (no other values having prefix '0101")

Main advantages of Vtrie over BST

Vtrie

- The complexity of looking up a key of length m is O(m).
- Tries can require less space when they contain a large number of short strings because the keys are not stored explicitly and nodes are shared between keys with common prefix.

Binary Search Tree

- The complexity of looking up a key of length m is O(mlogn).
- BST instead stores actual keys in nodes and nodes are not shared but independent.

Fatality of B+ tree on in-memory databases

- An enormous number of internal nodes.
- Node size is humongous.
- Hard implementation.

No bushy, no fat-ass trees!!!

Recovery-Checkpoint

- Checkpoint: SolidDB executes a snapshot-consistent checkpoint that is alone sufficient to recover the database to a consistent state that existed at some point in the past.
- SolidDB allows transaction logging to be turned off, if desired.

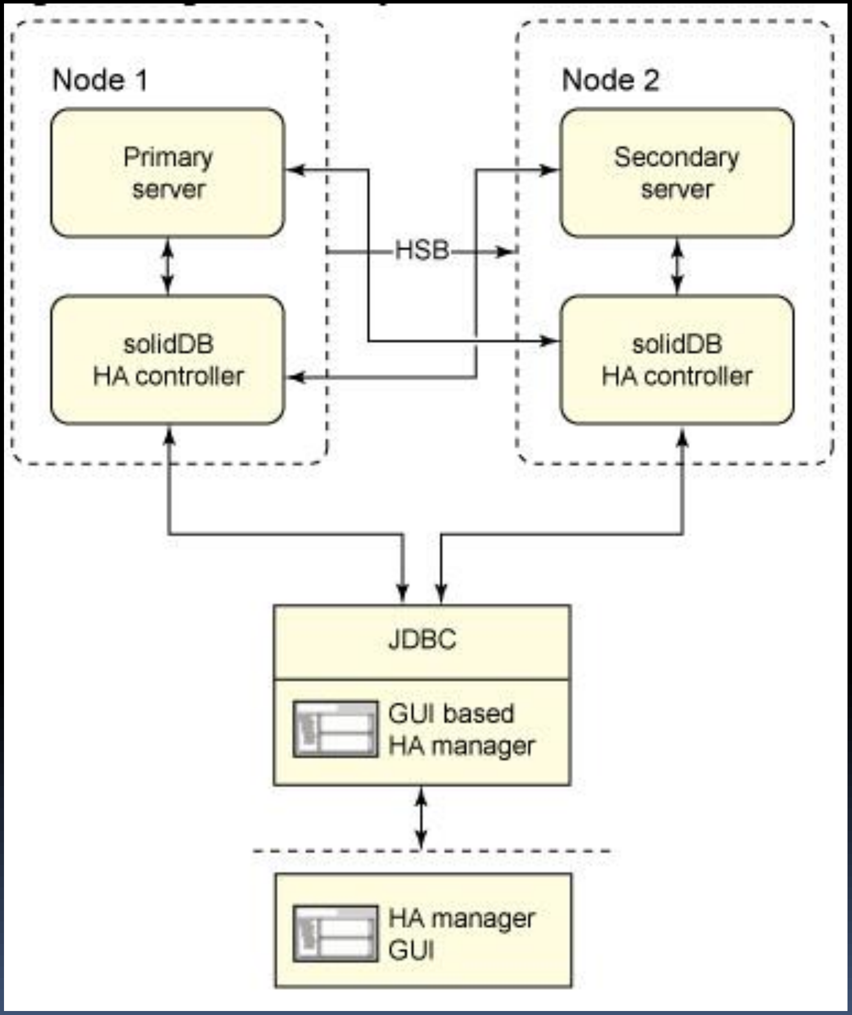
Availability:

High Availability Architecture:

1. Hot-Stanby (HSB) enables a secondary server to run in parallel with the primary server and keep an up-to-date copy of the data in the primary server.

2. **HA Controller** (HAC) is the automatic redundancy management program for IBM solidDB HSB. HAC detects failures, performs failovers, and restarts servers when necessary. HAC also has an API that enables HA Managers to connect to it.

3. **HA Manager** is a GUI-based tool that shows the status of HotStandby servers and the state of HA Controllers. The HA Manager also Includes basic functionality for managing the HAC. This tool is used in the demonstration to simulate a failure on the primary server and make a switch to the secondary server.



High availability-cont

- The main benefit of High Availability in an IBM solidDB environment is that applications are shielded from the effects of a failure of the primary database.
- Replication protocols (log writing): Synchronous (2Safe) and asynchronous (1Safe).

Distinctions between 2Safe and 1Safe

1Safe:

- It prefers performance over safety.
- It commits immediately without waiting for secondary’s response.

2Safe:

- It prefers safety over performance.
- 2Safe Received commits as soon as Secondary acknowledges that it has received transaction log.
- 2Safe Visible and 2Safe Committed both commit when Secondary has executed and committed the transaction.

Concurrency

- **Pessimistic concurrency control**
- **Optimistic concurrency control**

1. The solidDB implementation of optimistic concurrency control uses multiversioning.
2. If the version numbers are the same, then no one else changed the record and the system can write the updated value.
3. If the originally read value and the current value on the disk are not the same, then someone has changed the data

Conclusion

Authors conclude that solidDB has shown its trength on various business areas with low-latency and high throughput by comparing the performance between solidDB and a disk-based database, and giving the results of an experiment called Telecom Application Transaction Processing (TATP).

Unbeatable!