Shadow Storage

Idea: keep two versions for each **modified** block:

- old, consistent version
- new, (possibly) inconsistent version

atomic switch to indicate consistent versions

but: **two mapping tables**
Example: Modifying Transaction T1

version a

file

pointer to consistent version
Insert and Update

version a

file

version b

pointer to consistent version
Insert and Update

version a

file

version b

pointer to consistent version
Crash

version a

file

version b

M_a

0 1 2 3 4 5

B0 B1 B2 B3 B4 B5 B6 B7 B8 B9

M_b

0 1 2 3 4 5

pointer to consistent version

a
Persisting Changes

version a

file

version b

1. write modified blocks
2. write $M_b$
3. perform global switch

pointer to consistent version
Discussion

Advantages:

- Doubles storage only for changed blocks
- Undo of changes easy

Disadvantages:

- Helper data structures (maps) may become "big" (> 1 block)
- High degree of fragmentation

\( \text{virtual memory} \)

\( \text{ZFS} \)