

slides online at [http://web.mit.edu/6.033/www/assignments/lec17.\[ppt|pdf\]](http://web.mit.edu/6.033/www/assignments/lec17.[ppt|pdf])

Logging

6.033 Spring 2010 Lecture 17

Sam Madden

Key concepts:

Logging for all-or-nothing atomicity

Write ahead logging

Checkpoints

Shadow Copy Commit

//in memory copy of DB, some records dirty

commit(DB):

 copy(DB, DBnew)

 for each record r in memory:

 if r is dirty:

 write r into DBnew

 rename(DBnew, DB) //commit point!

 // rename must be atomic!

Read with just a log

```
// log is array of records with (type,TID,var,before,after)
read(var, log):
    cset = { }
    for r in log[len-1] ... log[0]:
        if (r.type == commit):
            cset = cset U r.TID
        if (r.type == update):
            if (r.TID in cset and r.var == var):
                return r.after
```

Read & write with cell storage

read(var):

return read(cell-storage, var)

write(TID, var, newval):

append(log, TID, var, read(var), newval)

write(cell-storage, var, newval)

Recover with cell-storage and log

```
recover(log):
```

```
  doneset = { }
```

```
  for each record r in log[len-1] ... log[0]: //UNDO
```

```
    if r.type == commit or r.type == abort
```

```
      doneset = doneset U r.TID
```

```
    if r.type == update and r.TID not in doneset:
```

```
      write(cell-storage, r.var, r.before)
```

Read with cache & cell-storage

read(var):

if var in cache:

return cache[var]

else:

val = read(cell-storage, var)

add (var,val) to cache // possibly evict other
// cached values

return val

write(var,val):

append(log,TID, var, cell-storage(var), newval)

add (var,val) to cache

Recover with cell-storage, log, and cache

```
recover(log):
```

```
  doneset = { }
```

```
  for each record r in log[len-1] ... log[0]: //UNDO
```

```
    if r.type == commit or r.type == abort
```

```
      doneset = doneset U r.TID
```

```
    if r.type == update and r.TID not in doneset:
```

```
      write(cell-storage, r.var, r.before)
```

```
  for each record r in log[0]...log[len-1]: //REDO
```

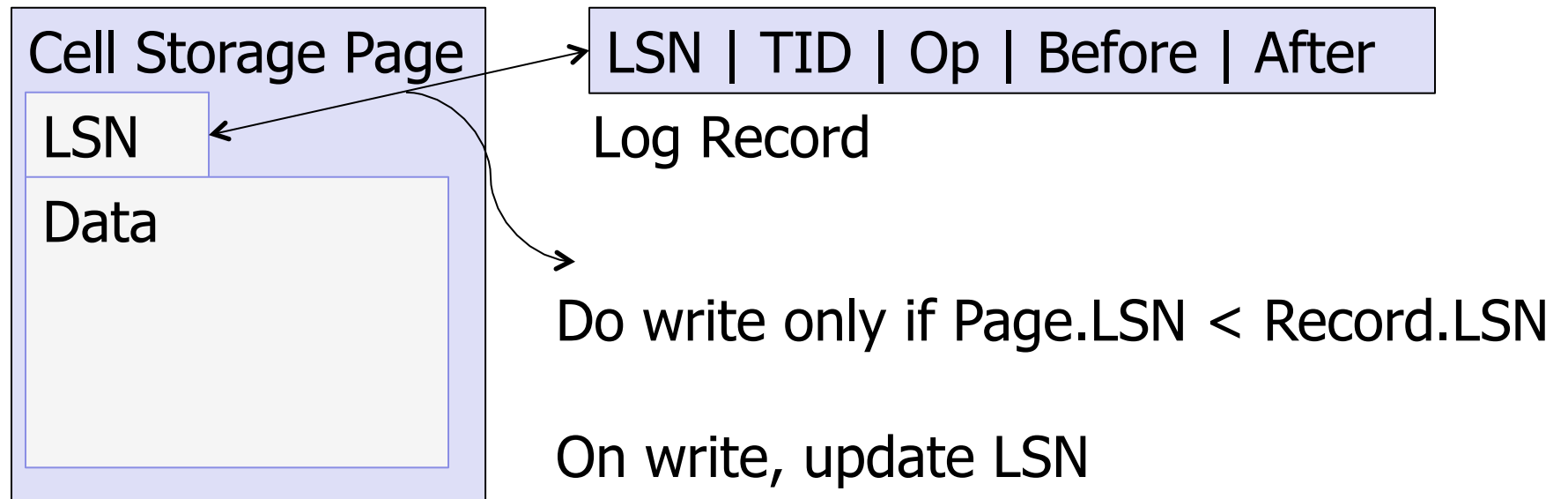
```
    if r.type == update and r.TID in doneset:
```

```
      write(cell-storage, r.var, r.after)
```

REDO Optimization

```
for each record r in log[0]...log[len-1]: //REDO
  if r.type == update and r.TID in doneset:
    if (cell-storage does not reflect r)
      write(cell-storage, r.var, r.after)
```

Log Sequence Number (LSN) is a unique increasing identifier for log records



FEEL SCREWED?



Big screw
April 5–9
Lobby 10

VOTE MADDEN