Állapotmentés és helyreállítás: Példák megvalósításokra

Majzik István

majzik@mit.bme.hu

Felhasználói állapotmentés memóriából fájlba C nyelvű, alacsony szintű megoldással

Checkpoint definíció

- Checkpoint vektor:

```
struct Cpt_vec {
    char *base;
    int len;
} cpt_vec[MAX_VARS];
int cpt_cnt;
```

- Checkpoint állomány:

```
static FILE *cpt_fd;
```

Mentendő adatok kijelölése

- Elmentendő program adatok (példa):

```
int i;
int results[DIM][DIM];
```

- Checkpoint vektor feltöltése:

```
cpt_cnt = 2;
cpt_vec[0].base = (char *)results;
cpt_vec[0].len  = sizeof(int [DIM][DIM]);
cpt_vec[1].base = (char *)&i;
cpt_vec[1].len  = sizeof(int);
```

Checkpoint adatok kimentése és betöltése

- Checkpoint adatok kimentése

```
fwrite((char *)&cpt_cnt,
        sizeof(int),
        1, cpt_fd);
fwrite((char *)cpt_vec,
        cpt_cnt*sizeof(struct Cpt_vec),
        1, cpt_fd);
for (i=0; i<cpt_cnt; i++) {
   fwrite(cpt_vec[i].base,
          cpt_vec[i].len,
          1, cpt_fd);
}
```

- Checkpoint adatok betöltése:
  - Hasonlóan, `fread()` hívással

Processzorállapot mentése és visszaállítása (Unix)

- A **`setjmp()`** és **`longjmp()`** rendszerhívások használhatók: „általános goto”
  - **`int setjmp(jmp_buf env);`** Állapot „rögzítés”
  - **`void longjmp(jmp_buf env, int val);`** Állapotba „ugrás”
- A **`longjmp()`** után a végrehajtás úgy folytatódik, mintha a hozzá tartozó **`setjmp()`** hívásból térne vissza
  - Ez esetben a **`longjmp()`** hívás második paramétere lesz a visszatérési érték
  - Rögzítés és visszaállítás megkülönböztethetők **`setjmp()`** esetén:
    - Sikeres állapot rögzítés: 0 értékkel tér vissza
    - Visszatöltés esetén: a beállított val értékkel tér vissza
- Hasonló funkcionalitás (signal mask mentésével):
  - **`sigsetjmp(),siglongjmp()`**

Processzorállapot mentése és visszaállítása (Unix)

- Belső állapot rögzítése:

```
jmp_buf st;
setjmp(st);
```

- A rögzített állapot mentése:

```
fwrite((char *)st,
        sizeof(jmp_buf), 1, cpt_fd);
```

- A rögzített állapot betöltése:

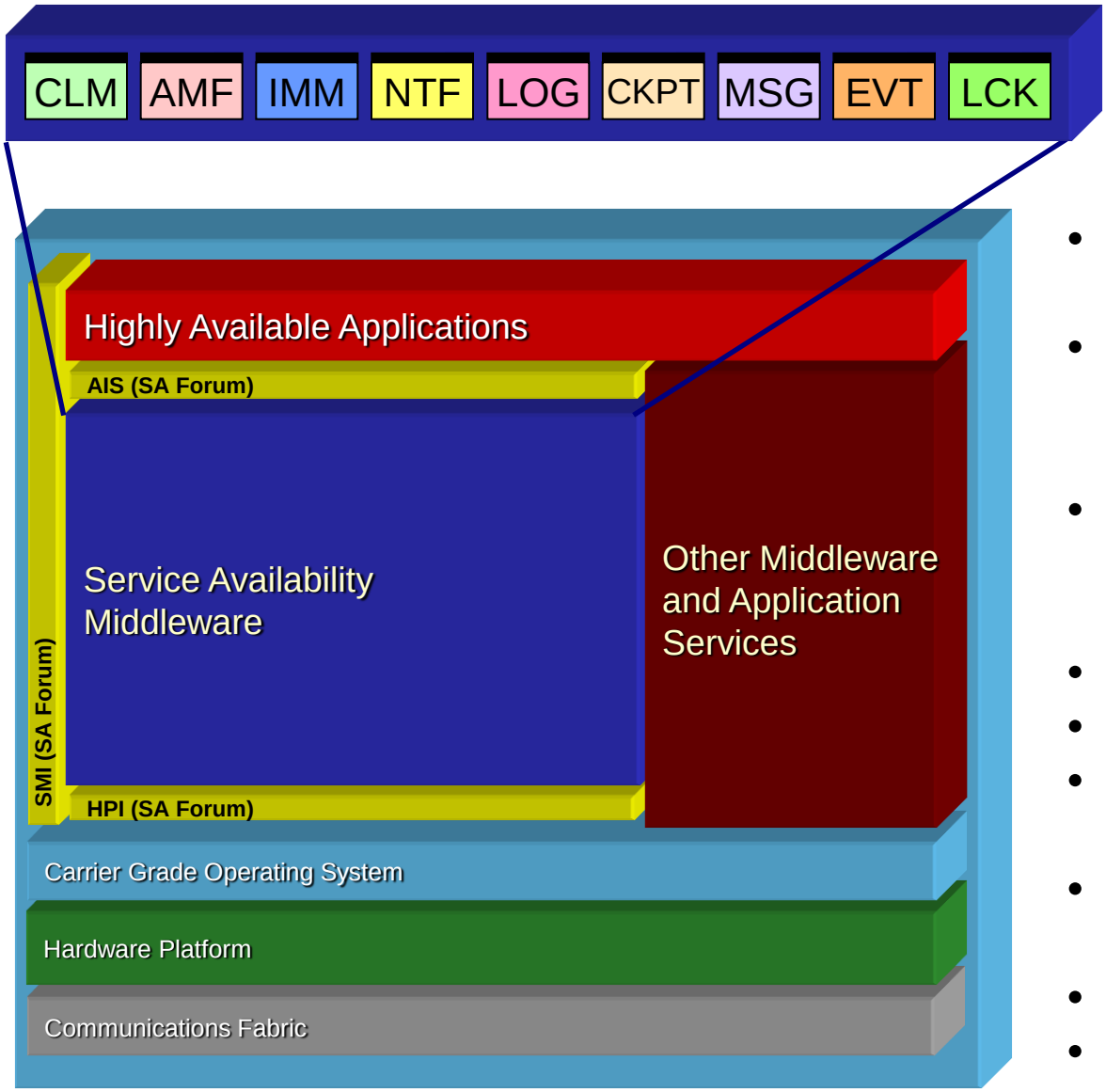
```
fread((char *)st,
        sizeof(jmp_buf), 1, cpt_fd);
```

- Visszalépés a mentett állapotba:

```
longjmp(st, 2);
```

Állapotmentés az SA Forum AIS köztesrétegben (C implementáció)

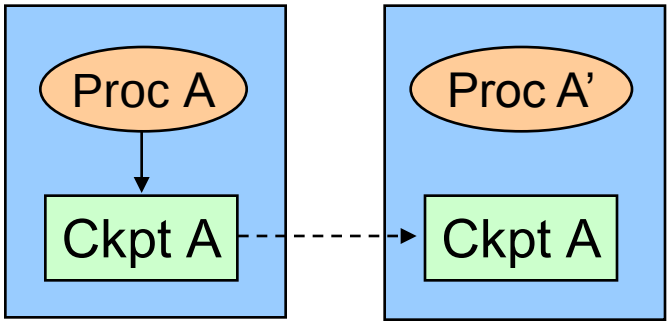
SA Forum köztesréteg szolgáltatások



- Cluster Membership Service (CLM)
- Availability Management Framework (AMF)
- Information Management Model (IMM)
- Notification (NTF)
- Log Service (LOG)
- Checkpoint Service (CKPT)
- Message Service (MSG)
- Event Service (EVT)
- Lock Service (LCK)

Az AIS köztesréteg checkpoint szolgáltatása

- Checkpoint adatok a fürt szervereinek memóriájában
  - Replikált checkpoint
- Konzisztencia biztosítása
  - Szinkron update
  - Aszinkron update
- Atomi hozzáférés garantálása
- Replikák menedzselése
  - Collocated checkpoint: Elsődleges kijelölése és replika létrehozás
  - Non-collocated checkpoint: Köztesréteg menedzseli a létrehozást
  - Érvényességi idő lejárta után törlés
- Hívások:
  - SaCkptCheckpointOpen, …Close, …Unlink
  - SaCkptCheckpointWrite, …Read
    - IoVector: Specifikálja a mentendő/helyreállítandó memóriatartományt
    - NumberOfElements: Bejegyzések száma az IoVector-ban
    - Bejegyzések részei: sectionId, dataBuffer, dataSize, dataOffset



Állapotok (objektumok) mentése J2SE alkalmazásokban

Java alapú megoldások

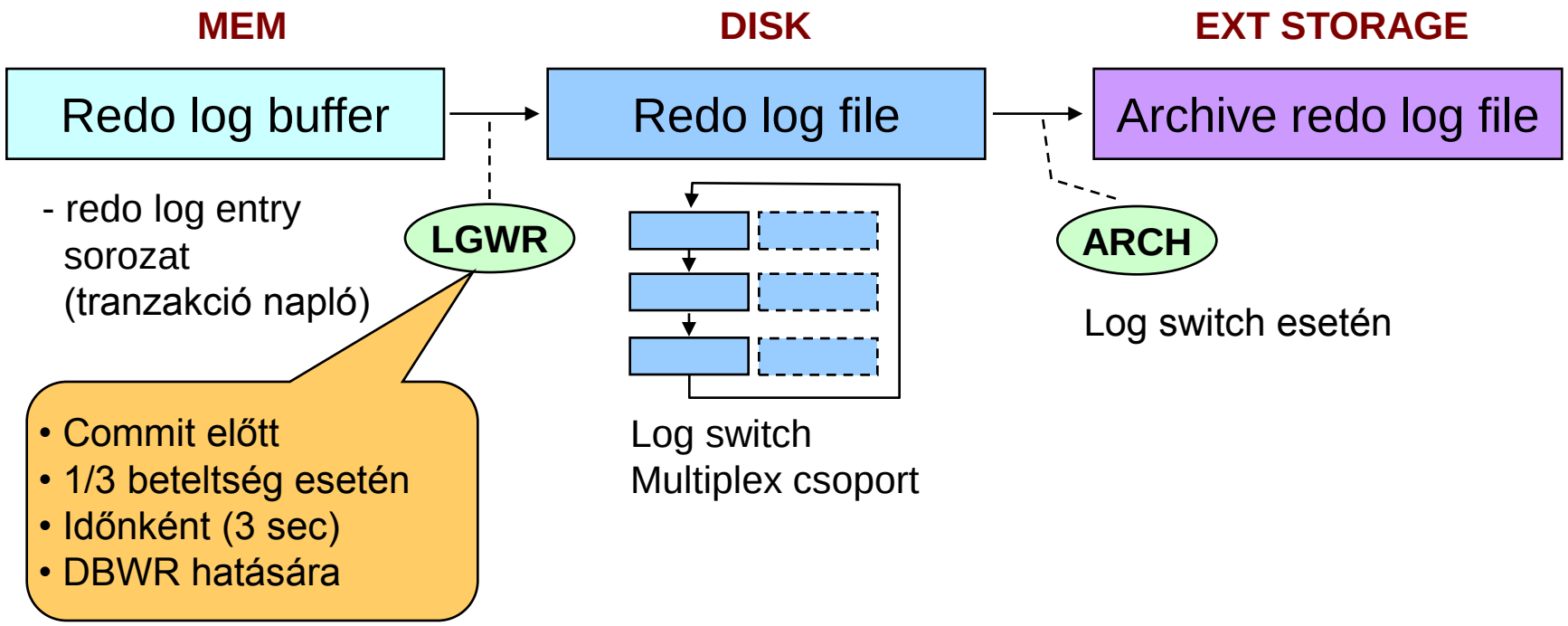
- Object serialization
  - Objektum gráfok tárolása binárisan (stream-ekben)
  - Serializable interfész, ObjectOutputStream writeObject
- Long-term persistence
  - Objektum gráfok tárolása XML fájlokban
  - XMLEncoder osztály writeObject metódus
- Java Database Connectivity (JDBC)
  - Objektumok explicit kiírása adatbázis táblába
  - SQL beágyazása Java forrásokba
- Java Data Objects (JDO)
  - Perzisztencia tulajdonságok megadása XML leíróban
  - Automatikus felműszerezés DB rutinokkal

Állapotmentés és helyreállítás adatbáziskezelő rendszerekben

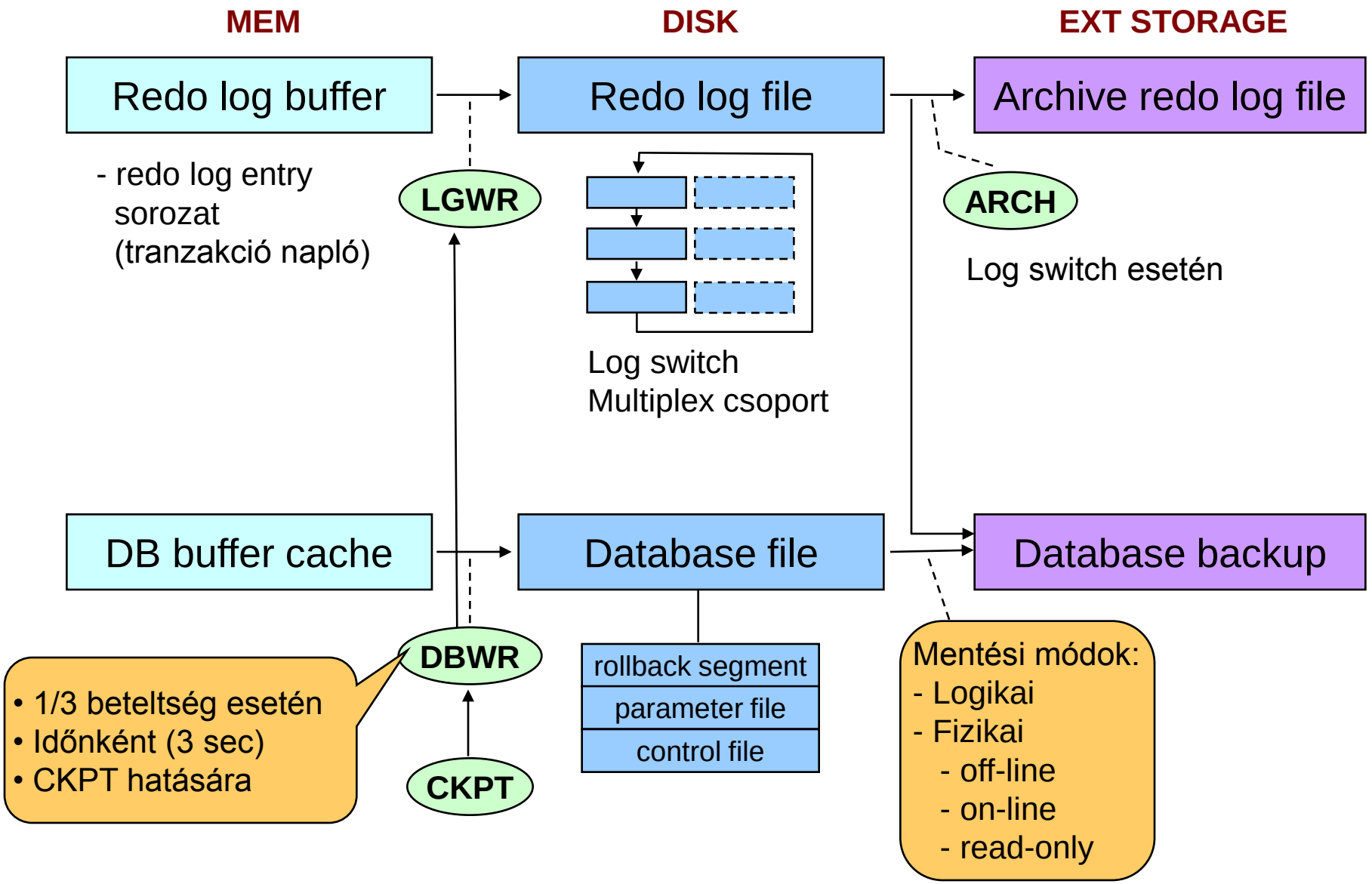
Mentések

- Mentendő adatok:
  - Tábla adatok: Adatbázis cache, adatbázis táblák
  - Rollback szegmensek
    - Abortált tranzakciók általi módosítások visszavonásához
  - Redo log
    - Műveletek naplózása újrajátszáshoz
  - Control információk
    - System area
    - User area
- Mentési hierarchia:
  - Mem $\rightarrow$ Disk $\rightarrow$ External (stable) storage

Mentési hierarchia I.



Mentési hierarchia II.



Helyreállítás

- Hibadetektálás:
  - SMON: rendszerfolyamatok felügyelete
    - SGA: System Global Area (locking, status, pool, …)
  - PMON: felhasználói folyamatok felügyelete
    - PGA: Program Global Area (stack, session memory)
  - Operátor
- Lépések a helyreállításhoz:
  - Adatbázis táblák visszatöltése a Database backup alapján
  - Roll forward (redo log alapján)
  - Rollback (rollback szegmensek alapján, nem committált műveletek)
- Mitől függ a mentési stratégia?
  - Adat kritikusság (fizikai / logikai mentés)
  - Adatbázis használat (on-line / off-line mentés)
  - Adatbázis hozzáférés gyakoriság (fizikai off-line / logikai mentés)