**1. strategia di rimpiazzo NAIVE con blocchi di 100 byte e 100 pagine di buffer**

* **dimensione di un record di R:** RL= 4 byte (used flag) + 4 byte (A intero) + 15 byte (B stringa) + 4 byte (intero per lunghezza stringa) = 27 byte
* Blocchi scritti per l’inserimento dei primi R=100.000 record

Consideriamo che in SimpleDB la memorizzazione è di tipo unspanned, quindi F = ⌊B/RL⌋ = ⌊100 / 27⌋ = 3 record

Da cui il numero di blocchi necessari per la memorizzazione è ⌈R/F⌉ = ⌈100.000 / 3⌉ = 3334 blocchi

Verranno quindi scritti 3334-1 = 3333 blocchi: ad ogni unpin si scarica una pagina, quindi tutte tranne l’ultima.

* **dimensione di un record di S:** RL= 4 byte (used flag) + 4 byte (C intero) + 4 byte (D intero) + 4 byte (E intero) = 16 byte
* Blocchi scritti per l’inserimento dei primi R=80.000 record

F = ⌊B/RL⌋ = ⌊100 / 16⌋ = 6 record

Da cui il numero di blocchi necessari per la memorizzazione è ⌈R/F⌉ = ⌈80.000 / 6⌉ = 1334 blocchi

Verranno quindi scritti 1334-1 = 1333 blocchi: ad ogni unpin si scarica una pagina, quindi tutte tranne l’ultima.

* Blocchi letti per la lettura delle ennuple di R per cui 1<=A<=1000, cioè di R = 1000 record

I record sono memorizzati in ⌈R/F⌉ = ⌈1000 / 3⌉ = 334 blocchi

Vengono letti 334 blocchi e scritto un blocco poiché occorre salvare (flush) anche l’ultima pagina utilizzata per l’inserimento precedente

* Blocchi scritti per la lettura delle ennuple di R per cui 2000<=A<=3000, cioè di R = 1001 record a cui vanno sommati I primi 2000.

I record sono memorizzati in ⌈R/F⌉ = ⌈3001 / 3⌉ = 1001 blocchi

Vengono letti 1001 blocchi

* Blocchi scritti per la lettura delle ennuple di R per cui 2500<=A<=2600, cioè di R = 101 record a cui vanno sommati I primi 2500.

I record sono memorizzati in ⌈R/F⌉ = ⌈2601 / 3⌉ = 867 blocchi

Vengono letti 867 blocchi

* Blocchi letti per la lettura delle ennuple di S per cui 1<=D<=1000, cioè di R = 1000 record

I record sono memorizzati in ⌈R/F⌉ = ⌈1000 / 6⌉ = 167 blocchi

Vengono letti 167 blocchi e scritto un blocco poiché occorre salvare (flush) anche l’ultima pagina utilizzata per l’inserimento precedente

* Blocchi letti per contare le ennuple di S per cui D=E, in questo caso è necessario leggere tutti I record.

I record sono memorizzati in ⌈R/F⌉ = ⌈80.000 / 6⌉ = 1334 blocchi

Vengono letti 1334 blocchi

**2. strategia di rimpiazzo NAIVE con blocchi di 100 byte e 1000 pagine di buffer**

* **dimensione di un record di R:** RL= 4 byte (used flag) + 4 byte (A intero) + 15 byte (B stringa) + 4 byte (intero per lunghezza stringa) = 27 byte
* Blocchi scritti per l’inserimento dei primi R=100.000 record

Consideriamo che in SimpleDB la memorizzazione è di tipo unspanned, quindi F = ⌊B/RL⌋ = ⌊100 / 27⌋ = 3 record

Da cui il numero di blocchi necessari per la memorizzazione è ⌈R/F⌉ = ⌈100.000 / 3⌉ = 3334 blocchi

Verranno quindi scritti 3334-1 = 3333 blocchi: ad ogni unpin si scarica una pagina, quindi tutte tranne l’ultima.

* **dimensione di un record di S:** RL= 4 byte (used flag) + 4 byte (C intero) + 4 byte (D intero) + 4 byte (E intero) = 16 byte
* Blocchi scritti per l’inserimento dei primi R=80.000 record

F = ⌊B/RL⌋ = ⌊100 / 16⌋ = 6 record

Da cui il numero di blocchi necessari per la memorizzazione è ⌈R/F⌉ = ⌈80.000 / 6⌉ = 1334 blocchi

Verranno quindi scritti 1334-1 = 1333 blocchi: ad ogni unpin si scarica una pagina, quindi tutte tranne l’ultima.

* Blocchi letti per la lettura delle ennuple di R per cui 1<=A<=1000, cioè di R = 1000 record

I record sono memorizzati in ⌈R/F⌉ = ⌈1000 / 3⌉ = 334 blocchi

Vengono letti 334 blocchi e scritto un blocco poiché occorre salvare (flush) anche l’ultima pagina utilizzata per l’inserimento precedente

* Blocchi scritti per la lettura delle ennuple di R per cui 2000<=A<=3000, cioè di R = 1001 record a cui vanno sommati I primi 2000.

I record sono memorizzati in ⌈R/F⌉ = ⌈3001 / 3⌉ = 1001 blocchi

Vengono letti 1001 blocchi

* Blocchi scritti per la lettura delle ennuple di R per cui 2500<=A<=2600, cioè di R = 101 record a cui vanno sommati I primi 2500.

I record sono memorizzati in ⌈R/F⌉ = ⌈2601 / 3⌉ = 867 blocchi

Vengono letti 867 blocchi

* Blocchi letti per la lettura delle ennuple di S per cui 1<=D<=1000, cioè di R = 1000 record

I record sono memorizzati in ⌈R/F⌉ = ⌈1000 / 6⌉ = 167 blocchi

Vengono letti 167 blocchi e scritto un blocco poiché occorre salvare (flush) anche l’ultima pagina utilizzata per l’inserimento precedente

* Blocchi letti per contare le ennuple di S per cui D=E, in questo caso è necessario leggere tutti I record.

I record sono memorizzati in ⌈R/F⌉ = ⌈80.000 / 6⌉ = 1334 blocchi

Vengono letti 1334 blocchi

3. strategia di rimpiazzo LRU con blocchi di 1000 byte e 250 pagine di buffer

* **dimensione di un record di R:** RL= 4 byte (used flag) + 4 byte (A intero) + 15 byte (B stringa) + 4 byte (intero per lunghezza stringa) = 27 byte
* Blocchi scritti per l’inserimento dei primi R=100.000 record

Consideriamo che in SimpleDB la memorizzazione è di tipo unspanned, quindi F = ⌊B/RL⌋ = ⌊1000 / 27⌋ = 37 record

Da cui il numero di blocchi necessari per la memorizzazione è ⌈R/F⌉ = ⌈100.000 / 37⌉ = 2703 blocchi

Verranno quindi scritti 2703-250 = 2453 blocchi: gli ultimi 250 blocchi restano sul buffer e non vengono scritti.

* **dimensione di un record di S:** RL= 4 byte (used flag) + 4 byte (C intero) + 4 byte (D intero) + 4 byte (E intero) = 16 byte
* Blocchi scritti per l’inserimento dei primi R=80.000 record

F = ⌊B/RL⌋ = ⌊1000 / 16⌋ = 62 record

Da cui il numero di blocchi necessari per la memorizzazione è ⌈R/F⌉ = ⌈80.000 / 62⌉ = 1291 blocchi

Verranno quindi scritti 1291 blocchi di cui 249 sono quelli di R che non erano stati scritti in precedenza, mentre altri 249 di S restano sul buffer.

* Blocchi letti per la lettura delle ennuple di R per cui 1<=A<=1000, cioè di R = 1000 record

I record sono memorizzati in ⌈R/F⌉ = ⌈1000 / 37⌉ = 28 blocchi

Vengono letti 28 blocchi e scritti altrettanti per liberare lo spazio sul buffer.

* Blocchi scritti per la lettura delle ennuple di R per cui 2000<=A<=3000, cioè di R = 1001 record a cui vanno sommati altri 1000 poiché I primi 1000 sono già salvati sul buffer.

I record sono memorizzati in ⌈R/F⌉ = ⌈2001 / 37⌉ = 54 blocchi

Vengono letti 54 blocchi e scritti altrettanti per liberare lo spazio sul buffer.

* Blocchi scritti per la lettura delle ennuple di R per cui 2500<=A<=2600, cioè di R = 101 record.

Questi record si trovano già sul buffer e dunque non viene letto né scritto nulla sulla momoria principale

* Blocchi letti per la lettura delle ennuple di S per cui 1<=D<=1000, cioè di R = 1000 record

I record sono memorizzati in ⌈R/F⌉ = ⌈1000 / 62⌉ = 17 blocchi

Vengono letti 17 blocchi e scritti scritti altrettanti per liberare lo spazio sul buffer.

* Blocchi letti per contare le ennuple di S per cui D=E, in questo caso è necessario leggere tutti I record.

I record sono memorizzati in ⌈R/F⌉ = ⌈80.000 / 62⌉ = 1291 blocchi

Vengono letti 1291-17=1274 blocchi poiché 17 si trovano già sul buffer.

4. strategia di rimpiazzo LRU con blocchi di 500 byte e 10.000 pagine di buffer

* **dimensione di un record di R:** RL= 4 byte (used flag) + 4 byte (A intero) + 15 byte (B stringa) + 4 byte (intero per lunghezza stringa) = 27 byte
* Blocchi scritti per l’inserimento dei primi R=100.000 record

Consideriamo che in SimpleDB la memorizzazione è di tipo unspanned, quindi F = ⌊B/RL⌋ = ⌊500 / 27⌋ = 18 record

Da cui il numero di blocchi necessari per la memorizzazione è ⌈R/F⌉ = ⌈100.000 / 18⌉ = 5556 blocchi

Non verrà scritto alcun blocco poiché il buffer è sufficientemente grande da contenerli tutti.

* **dimensione di un record di S:** RL= 4 byte (used flag) + 4 byte (C intero) + 4 byte (D intero) + 4 byte (E intero) = 16 byte
* Blocchi scritti per l’inserimento dei primi R=80.000 record

F = ⌊B/RL⌋ = ⌊500 / 16⌋ = 31 record

Da cui il numero di blocchi necessari per la memorizzazione è ⌈R/F⌉ = ⌈80.000 / 31⌉ = 2581 blocchi

Anche in questo caso non viene scritto nulla poiché 2581+5556 < 10.000, dunque il buffer è in grado di contenere in memoria entrambe le relazioni.

* Poiché le relazioni sono tutte sul buffer per le richieste al punto B non verrà letto nulla dalla memoria centrale, le operazioni verranno svolte solamente sul buffer.