

---

# **Esercitazione E7**

## **Prova di autovalutazione**

# Esercizio 1

---

Un file di 8800 record di 180 Byte (con chiave da 47 Byte), è allocato su di un disco, denominato A, con cluster di allocazione da 512 Byte, e su di esso è costruito un indice ISAM. Il file viene copiato su un secondo disco, denominato B, con cluster di allocazione da 4 KB, e anche sul nuovo disco viene costruito l'indice ISAM. Si supponga che entrambi i dischi abbiano cilindri da 10 MB, indirizzi fisici dei blocchi di 4 Byte, tempo di accesso (seek+latency) pari a 20 ms e velocità di trasferimento di 1 MB/s, e che mai un record possa essere spezzato su due cluster.

Calcolare:

- a) l'occupazione totale in KByte del file e dell'indice in entrambe le organizzazioni;
- b) la percentuale di spazio inutilizzato nei blocchi del file dati e del file indice in entrambe le organizzazioni;
- c) il tempo complessivo necessario nell'organizzazione A ad accedere ad un record di chiave data **usando l'indice** e supponendo che sia il file che l'indice siano *completamente frammentati*;
- d) il tempo complessivo necessario nell'organizzazione B ad accedere ad un record di chiave data *senza usare l'indice* e supponendo che il file sia completamente deframmentato.

# Occupazione dei file nel disco A

---

- a) l'occupazione totale in Kbyte del file e dell'indice in entrambe le organizzazioni;
- Ciascun blocco del file dati contiene  $R^A = \lfloor B^A / r \rfloor = \lfloor 512 / 180 \rfloor = 2$  record/blk. Quindi il file dati è costituito da  $F^A = \lceil n / R^A \rceil = \lceil 88.000 / 2 \rceil = 44.000$  blocchi, e occupa  $44.000 \cdot 512 \text{ Byte} = 22.000 \text{ Kbyte} \approx 22 \text{ MB}$ .
  - I record del file indice sono in numero pari ai blocchi del file dati, cioè:  $F^A = 44.000$ . Ciascuno di essi è costituito da  $(c+b) = 47+4 = 51$  byte.
  - Ciascun blocco del file indice contiene  $R^A_i = \lfloor B^A / (c+b) \rfloor = \lfloor 512 / 51 \rfloor = 10$  rec/blk. Quindi il file indice è costituito da  $I^A = \lceil F^A / R^A_i \rceil = \lceil 44.000 / 10 \rceil = 4400$  blocchi, e occupa  $4400 \cdot 512 \text{ Byte} = 2200 \text{ Kbyte}$

# Occupazione dei file nel disco B

---

a) l'occupazione totale in Kbyte del file e dell'indice in entrambe le organizzazioni;

- Ciascun blocco del file dati contiene  $R^B = \lfloor B^B / r \rfloor = \lfloor 4096 / 180 \rfloor = 22$  record/blk. Quindi il file dati è costituito da  $F^B = \lceil n / R^B \rceil = \lceil 88.000 / 22 \rceil = 4.000$  blocchi, e occupa  $4.000 \cdot 4 \text{ KByte} = 16.000 \text{ Kbyte} \approx 16 \text{ MB}$ .
- I record del file indice sono in numero pari ai blocchi del file dati, cioè:  $F^B = 4.000$ . Ciascuno di essi è costituito da  $(c+b) = 47+4 = 51$  byte.
- Ciascun blocco del file indice contiene  $R^B_i = \lfloor B^B / (c+b) \rfloor = \lfloor 4096 / 51 \rfloor = 80$  rec/blk. Quindi il file indice è costituito da  $I^B = \lceil F / R^B_i \rceil = \lceil 4.000 / 80 \rceil = 50$  blocchi, e occupa  $50 \cdot 4 \text{ KByte} = 200 \text{ Kbyte}$ .

# Spazio inutilizzato nel disco A

---

b) la percentuale di spazio inutilizzato nei blocchi del file dati e del file indice in entrambe le organizzazioni.

– Ciascun blocco del file dati contiene  $R^A = 2$  record.

– Pertanto sono utilizzati  $R^A \cdot r = 2 \cdot 180 = 360$  byte. Ne restano quindi inutilizzati  $B^A - R^A \cdot r = 512 - 360 = 152$  byte.

– La frazione di spazio inutilizzata nei blocchi del file dati è quindi:

$$(B^A - R^A \cdot r) / B^A = 152 / 512 \approx 0.3 = 30\%$$

– Ciascun blocco del file indice contiene  $R^A_i = 10$  record.

– Pertanto sono utilizzati  $R^A_i \cdot (c+b) = 10 \cdot 51 = 510$  byte. Ne restano quindi inutilizzati  $B^A - R^A_i \cdot (c+b) = 512 - 510 = 2$  byte.

– La frazione di spazio inutilizzata nei blocchi del file dati è quindi:

$$(B^A - R^A_i \cdot (c+b)) / B^A = 2 / 512 \approx 0.04 = 0.4\%$$

# Spazio inutilizzato nel disco B

---

- b) la percentuale di spazio inutilizzato nei blocchi del file dati e del file indice in entrambe le organizzazioni.
- Ciascun blocco del file dati contiene  $R^B = 22$  record.
  - Pertanto sono utilizzati  $R^B \cdot r = 22 \cdot 180 = 3960$  byte. Ne restano quindi inutilizzati  $B^B - R^B \cdot r = 4096 - 3960 = 136$  byte.
  - La frazione di spazio inutilizzata nei blocchi del file dati è quindi:  
$$(B^B - R^B \cdot r) / B^B = 136 / 4096 \approx 0.03 = 3\%$$
  - Ciascun blocco del file indice contiene  $R^{B_i} = 80$  record.
  - Pertanto sono utilizzati  $R^{B_i} \cdot (c+b) = 80 \cdot 51 = 4080$  byte. Ne restano quindi inutilizzati  $B^B - R^{B_i} \cdot (c+b) = 4096 - 4080 = 16$  byte.
  - La frazione di spazio inutilizzata nei blocchi del file dati è quindi:  
$$(B^B - R^{B_i} \cdot (c+b)) / B^B = 16 / 4096 \approx 0.04 = 0.4\%$$

# Tempi di accesso al disco

---

- Per rispondere ai quesiti c) e d) occorre calcolare i tempi di accesso e di trasferimento
- Il tempo di accesso è fornito dall'enunciato:  **$T_a = 20\text{ms}$** .
- La velocità di trasferimento è anch'essa fornita dall'enunciato ed è di  **$1 \text{ MB/s} = 2^{20} \text{ byte/s}$**
- I blocchi del disco **A** richiedono un tempo di trasferimento pari a  **$T_t^A = 2^9 / 2^{20} = 2^{-11} \text{ s} \approx 1 / 2000 \text{ s} = 0.5 \text{ ms}$**
- I blocchi del disco **B**, essendo 8 volte più grandi richiedono un tempo di trasferimento pari a  **$T_t^B = 8 T_t^A \approx 4 \text{ ms}$**

# Tempo di accesso nel disco A

---

c) il tempo complessivo necessario nell'organizzazione A ad accedere ad un record di chiave data **usando l'indice** e supponendo che sia il file che l'indice siano *completamente frammentati*;

- Essendo la dimensione del file indice pari a  $I^A = 4400$  blocchi, ed essendo il file completamente frammentato, occorrerà in media effettuare  $I^A / 2 + 1 = 2201$  accessi, e trasferire **2201** blocchi.
- Pertanto complessivamente il tempo medio di accesso sarà:

$$2201 \cdot (T_a + T_t^A) = 2201 \cdot 20.5 \text{ ms} = 45120.5 \text{ ms} = 45 \text{ s}$$

# Tempo di accesso nel disco B

---

d) il tempo complessivo necessario nell'organizzazione B ad accedere ad un record di chiave data *senza usare l'indice* e supponendo che il file sia completamente deframmentato

- Dato che non si utilizza l'indice, occorrerà in media effettuare accedere alla metà dei blocchi  $F^B / 2 = 4000 / 2 = 2000$ .
- Essendo il file completamente deframmentato sarà sufficiente un solo tempo di accesso e 2000 tempi di trasferimento
- Pertanto complessivamente il tempo medio di accesso sarà:

$$T_a + 2000 \cdot T_t^B = 20 \text{ ms} + 2000 \cdot 4 \text{ ms} = 8020 \text{ ms} = 8.2 \text{ s}$$

## Esercizio 2

---

Un video server gestisce stream a 30 frame. I frame sono registrati su disco in blocchi di 1 MB che contengono ciascuno 15 frame. Nella configurazione attuale il server è dotato di 4 CPU a 500 MHz e di 4 dischi con tempo di seek di 10 ms, rotazione a 3.000 giri/minuto e tracce da 4 MB. Con una delle CPU attuali per elaborare un frame occorre 1 ms. Supponendo che si desideri mantenere l'utilizzazione delle CPU e dei dischi rispettivamente al di sotto del 90% e del 40%:

- a)** determinare il massimo numero di stream  $S_{\max}$  che il sistema può gestire e le corrispondenti utilizzazioni di dischi e CPU;
- b)** calcolare come occorre aumentare il numero di dischi o di CPU in modo da bilanciare il sistema, e il corrispondente nuovo valore di  $S_{\max}$  ;
- c)** con riferimento alla configurazione ed al valore di  $S_{\max}$  calcolati al punto **b)**, calcolare in che misura può essere ridotto il numero di CPU e di dischi passando ad una configurazione con CPU da 2.5 GHz e organizzazione dei dischi con blocchi da 8 MB e 120 frame;
- d)** calcolare la quantità di memoria necessaria ad allocare doppi buffer di memoria per tutte le stream nelle configurazioni di cui ai punti **b)** e **c)**.

# Tempi di accesso ai blocchi

---

- Tempo di seek dall'enunciato:  $t_s = 10 \text{ ms}$
- Tempo di latency (metà del tempo di rotazione = 50 giri/s):  
$$t_{\text{lat}} = t_{\text{rot}} / 2 = (1/50) / 2 = 10 \text{ ms}$$
- Velocità di trasferimento  $V_T = \text{trk} / t_{\text{rot}} = 4 \text{ MB} / 20 \text{ ms} = 200 \text{ MB/s}$
- Tempo di trasferimento di un blocco  
$$t_t = 1 \text{ MB} / V_T = 1 \text{ MB} / 200 \text{ MB/s} = 1/200 \text{ s} = 5 \text{ ms}$$
- Complessivamente la lettura di un blocco comporta:  
$$t_B = t_s + t_{\text{lat}} + t_t = 25 \text{ ms}$$

# Max numero di stream gestibili dalle CPU

---

- Un frame richiede 1 ms su una CPU a 500 MHz
- L'utilizzazione delle 4 CPU attuali con  $s$  stream è data da:

$$U_{\text{CPU}} = (s \cdot 30 \cdot 0.001) / n_{\text{CPU}} = (s \cdot 0.030) / 4$$

- Volendo limitare l'utilizzazione delle CPU al 90%:

$$0.9 = (s \cdot 0.030) / 4$$

- Quindi il massimo numero di stream gestibile dalle CPU è:

$$S_{\text{max}}^c = 0.9 \cdot 4 / 0.030 = 120$$

# Max numero di stream gestibili dal disco

---

- Ciascuna stream richiede la lettura di **30** frame/s, quindi di due blocchi (da **15** frame) ogni secondo
- In totale una stream contribuisce all'utilizzazione del disco per:

$$\Delta U_{\text{DISK}} = t_B \cdot 2 = 25 \text{ ms} \cdot 2 = 0.050 \text{ s}$$

- L'utilizzazione dei 4 dischi attuali con **s** stream è data da:

$$U_D = (s \cdot \Delta U_{\text{DISK}}) / n_D = (s \cdot 0.050) / 4$$

- Volendo limitare l'utilizzazione dei dischi al 40%:

$$0.4 = (s \cdot 0.050) / 4$$

- Quindi il massimo numero di stream gestibile dai dischi è:

$$S^D_{\text{max}} = 0.4 \cdot 4 / 0.050 = 32$$

# Max numero di stream gestibili

---

- a) determinare il massimo numero di stream  $S_{\max}$  che il sistema può gestire e le corrispondenti utilizzazioni di dischi e CPU;
- I dischi costituiscono il collo di bottiglia, poiché raggiungono la loro utilizzazione massima (40%) già con 32 stream
  - Anche se le CPU potrebbero sostenere un carico superiore, il sistema può numero massimo di stream pari a  $S_{\max} = 32$
  - Con  $S_{\max} = 32$  l'utilizzazione dei dischi è del 40%
  - Con  $S_{\max} = 32$  l'utilizzazione dei dischi è:

$$U_{\text{CPU}} = (S_{\max} \cdot 0.030) / 4 = (32 \cdot 0.030) / 4 = 0.24 = 24\%$$

# Bilanciamento della configurazione

---

**b)** calcolare come occorre aumentare il numero di dischi o di CPU in modo da bilanciare il sistema, e il corrispondente nuovo valore di  $S_{\max}$  ;

- I dischi costituiscono il collo di bottiglia: quindi occorre aggiungere dischi per bilanciare la configurazione ed avere  $S_{\max} = S_{\max}^c = 120$
- Il numero dei dischi  $n_D$  deve crescere per portare  $S_{\max}^D$  a raggiungere  $S_{\max}^c$  pur mantenendo  $U_D$  al di sotto del 40%

$$0.4 \geq U_D = (S_{\max}^c \cdot \Delta U_{\text{DISK}}) / n_D$$

$$0.4 \geq (120 \cdot 0.050) / n_D$$

$$n_D \geq (120 \cdot 0.050) / 0.4 = 15$$

- Cioè bisogna aggiungere altri 11 dischi portando il totale a 15.

# Sostituzione di CPU e dischi

---

c) con riferimento alla configurazione ed al valore di  $S_{\max}$  calcolati al punto b), calcolare in che misura può essere ridotto il numero di CPU e di dischi passando ad una configurazione con CPU da 2.5 GHz e organizzazione dei dischi con blocchi da 8 MB e 120 frame;

- Con CPU da 2.5 GHz i tempi si abbattano di un fattore 5: pertanto basta un quinto delle CPU, cioè una sola CPU.

- Con la nuova dimensione di blocco, il tempo di trasferimento aumenta di un fattore 8: quindi  $t'_t = 40 \text{ ms}$ , e  $t'_B = 60 \text{ ms}$

- In compenso ogni stream richiede una lettura ogni 4 secondi e quindi  $\Delta U_{\text{DISK}} = t'_B / 4 = 15 \text{ ms}$

$$0.4 \geq U_D = (S_{\max} \cdot \Delta U_{\text{DISK}}) / n_D$$

$$0.4 \geq (120 \cdot 0.015) / n_D \Rightarrow n_D \geq (120 \cdot 0.015) / 0.4 = 4.5$$

- Bastano 5 dischi

# Buffer di memoria

---

d) calcolare la quantità di memoria necessaria ad allocare doppi buffer di memoria per tutte le stream nelle configurazioni di cui ai punti b) e c).

- Nella configurazione di cui al punto b) le stream sono  $S_{\max} = 120$  e i blocchi sono da **1 MB**:

$$M^b = S_{\max} \cdot 2 \cdot 1 \text{ MB} = 120 \cdot 2 \cdot 1 \text{ MB} = 240 \text{ MB}$$

- Nella configurazione di cui al punto c) le stream sono sempre  $S_{\max} = 120$  , ma i blocchi sono da **8 MB**:

$$M^c = S_{\max} \cdot 2 \cdot 8 \text{ MB} = 120 \cdot 2 \cdot 8 \text{ MB} = 1920 \text{ MB}$$

# Sicurezza e protezione

---

- F** - per verificare una firma digitale occorre disporre della chiave privata del firmatario;
- V** - gli attacchi passivi sono quelli tesi ad intercettare i dati ed a carpirne il contenuto;
- F** - i certificati digitali contengono la chiave privata del loro proprietario;
- V** - lo *spamming* viene usato negli attacchi di tipo *denial of service*;
- F** - gli attacchi attivi sono quelli tendenti a violare la confidenzialità dei dati;
- V** - lo *sniffing* viene usato in fase di realizzazione di attacchi passivi;
- F** - i certificati digitali sono firmati dai loro proprietari con la loro chiave pubblica;

# Processi e thread

---

- F** - i processi CPU-bound hanno CPU burst corti;
- V** - thread dello stesso processo possono avere, ad un dato istante, valori diversi per lo stesso registro;
- F** - lo scheduling *round-robin* della CPU provoca una sottoutilizzazione dei dispositivi di I/O;
- V** - i file aperti sono risorse condivise dai thread di uno stesso processo;
- V** - lo scheduling dei dischi con l'algoritmo dell'ascensore garantisce un limite all'attesa in coda;
- V** - nello scheduling di Windows 2000 i processi utente hanno sempre priorità inferiore a quelli di sistema
- V** - nello scheduling round-robin il tempo di attesa in coda `e proporzionale alla richiesta di servizio;

# Sincronizzazione e deadlock

---

- V** - il *rollback* costituisce un metodo per la risoluzione del deadlock;
- F** - la mutua esclusione è condizione sufficiente al verificarsi del deadlock;
- F** - l'uso dei semafori garantisce il programmatore nei confronti del deadlock;
- F** - due processi che non accedono mai a risorse comuni non possono mai essere coinvolti in una situazione di stallo;
- F** - il deadlock si può verificare anche quando una sola delle condizioni necessarie è soddisfatta;
- V** - l'attesa circolare è condizione sufficiente al verificarsi il deadlock;
- F** - nel produttore-consumatore il produttore deve porsi in attesa quando il buffer è vuoto;

# Gestione della memoria

---

- V** - il TLB serve a limitare la frequenza degli accessi alla tavola delle pagine;
- V** - a parità di altre condizioni la località provoca una diminuzione del *Page Fault Rate*;
- V** - nell'algoritmo del *working set* ad un allargamento della finestra consegue un aumento dei page frames assegnati ai processi;
- F** - località temporale significa forte probabilità di fare riferimento ad indirizzi contigui in tempi vicini;
- F** - lo *swapping* fu introdotto nell'ambito dei primi sistemi batch;
- F** - l'algoritmo di rimpiazzamento ottimo viene usato raramente per il fatto che è più oneroso degli altri;
- F** - l'organizzazione a più livelli delle tavole delle pagine serve a limitare la dimensione complessiva della tavola;