データベースシステム 88－2
（1992. 5. 8）

部分的主記憶データベースシステムにおける トランザクション処理能力の評価

高倉弘喜　上林彌彦
京都大学工学部

高速処理データベースシステム実現法の1つとして主記憶データベースが注目されつつある。本稿では主記憶容量の限界を考慮し、良く使われるホットスポットデータだけを主記憶データベース化し、非ホットスポットデータは従来と同じディスクデータベースとして扱う部分的主記憶データベースを提案する。また、TPC ベンチマーク B に基づくシミュレーションにより部分的主記憶データベースでも、データベース全体を主記憶データベース化したのとそれほど変わらない性能が期待できることを示す。

Performance Evaluation of Transaction Processing on a Partial Main Memory Database

Hiroki TAKAKURA　Yahiko KAMBAYASHI
Faculty of Engineering, Kyoto University
Yoshidahonmachi, Sakyou-Ku, Kyoto 606-01, Japan

Studies on main memory database systems have been getting popular, since they seems to be practical methods to realize high-performance database systems. There is, however, a problem caused by the size limit of main memory, as required data size is growing very rapidly. We propose a partial main memory database where frequently-used hotspot data are resided in the main memory and non-hotspot data are treated as a conventional disk database. From the simulation, we can expect that a partial main memory database has almost equivalent performance as a main memory system where whole databases are stored in the main memory.

1. まえがき

近年、データベースがさまざまな分野で利用されるようになり、データベースシステムに対しさらに高いデータ処理能力が要求されつつある。高い処理能力を実現するため、効率的な質問処理[KIM85]、並行処理制御[LEH89]、データベースマシーンや主記憶データベース[AMM85][DEW84][EIC88][EIC89][HAG86][KAM91][LEH86a][LEH86b][SHA86][SON87]などの研究や開発がこれまで議論されてきた。このうち主記憶データベースは効率的な障害対策をいかに実現するかという問題があるが、処理を遅らせる要因の一つであるディスクI/Oの影響を大幅に減らすことができる。

これまでの主記憶データベースの議論では、主記憶にデータベース全体を常駐させる完全主記憶データベースに関するものが主であった。完全主記憶データベースはディスクI/Oが全く存在しないので、1秒間に千トランザクション程度の処理を行なうような高性能が期待できると言われている。しかし、全データベースを主記憶に常駐させることは主記憶容量の物理的限界等から考えて現実的でない。

そこで本稿では、良く使われるデータのみを主記憶に常駐させ、そうでないデータは必要に応じてディスクから読み出す部分的主記憶データベースを提案し、TPCベンチマークBを想定したシミュレーションによる部分的主記憶データベースのトランザクション処理能力について述べる。部分的主記憶データベースは、完全主記憶データベースと比べ、ディスクI/Oが存在する分だけ性能が劣るが、将来要求されるトランザクション処理能力を実現できると考えられる。

本研究は我々の行なってきたトランスペアレントバックアップに関連して行なっているもので、主記憶のホットスポット部分のバックアップを効率良く行なえればバックアップの面でも主記憶データベースより有効である。

2. 基本的事項

ホットスポットデータ (80/20 ルール)

データベースに対するアクセスはデータベースのわずかな部分に集中する傾向がある。典型的なデータベースシステムでは、約80%のアクセスがデータベースの約20%の部分に集中する傾向があることが知られている (80/20 ルール)[GAW85]。このようなアクセスの集中するデータをホットスポットデータと呼ぶ。

部分的主記憶データベース

主記憶データベースは高性能システムを実現する方法の1つであると考えられる。完全主記憶データベースは全データベースを主記憶に常駐させる[EIC86][EIC87]ため、コストや物理的な問題から現実的でないと考えられる。

本稿では、80/20 ルールを考慮して、ホットスポットデータだけを主記憶に常駐させ、非ホットスポットデータは、従来のディスクデータベースと同様に、必要になったときに

ディスクから主記憶へ読み出す部分的主記憶データベースを提案する。部分的主記憶データベースは限られた主記憶容量のもとで高性能システムを実現するという問題を解決できると考えられる。なお、本稿では主記憶データベースの障害対策については述べない。

データ更新の仮定

本稿のシステムでは、更新されたページはデータベースから読み出した同じ場所(番地)に反映されるものとする(in-place 更新)。

3. シミュレーションの概要

3.1 TPC ベンチマーク B

TPC ベンチマーク B[GRA91] は銀行のオンラインシステムを想定し、各トランザクションは口座データベース Account、支店データベース Branch、出納係データベース Teller、履歴データベース Histoy の 4 つのデータベースにアクセスする。History を除いて、3 つのデータベースのサイズは 1 秒間に処理されるトランザクション数 (tps) に比例する。そこで、3 つのデータベースのサイズを $C \cdot tps$ と表す。ここで C は定数であり、少なくとも 10,001,100 バイト以上でなければならない。History は 8 時間の動作中に生成される全てのレコードを保存できるだけの容量を持たなければならない。各トランザクションは次のようなデータ更新を行う。

Begin transaction

Account のレコードを更新 (Account_balance=Account_balance+delta)

Teller のレコードを更新 (Teller_balance=Teller_balance+delta)

Branch のレコードを更新 (Branch_balance=Branch_balance+delta)

History にレコードを追加

Commit transaction

データベースの定義を表 1 に示す。

データベース	レコード長	フィールド	レコード数 /tps
Account	100B	Account-ID, Branch-ID, Account, balance	100000
Branch	100B	Branch-ID, Branch, balance	1
Teller	100B	Teller-ID, Branch-ID, Teller, balance	10
History	50B	Account-ID, Teller-ID, Branch-ID, delta, Timestamp	

表 1: データベースの定義

3.2 ディスクの仕様

現在および数年後に予想されるディスク技術の概要を表2に示す。また、表2から数年後に予想される3.5インチディスクの仕様を表3に示す。ここで、各ディスク装置は2枚のディスクを持つとする。従って、1装置の容量は約400Mバイトになる。本稿ではこのような次世代のディスクを考えることにするが、ディスクヘッドはボイスコイル方式で制御されているとし、シリンダ間のシーク時間は次式で求められる [STR83]。

$$Seek\_time = a + b\sqrt{i}$$

ここで、iはシリンダ間の距離を表し、$a = 5$[ms]、$b = 0.5$[ms] とする。

	現在	数年後
最大線記録密度 [BPI]	50000	70000
トラック密度 [TPI]	2000	3000
回転時間 [rpm]	3600	5000
最高シリアル・データ転送速度 [Mb/s]	15	35

表 2: ディスク技術の概要

シリンダ数	セクタ数	一面当たりの容量	ディスク枚数
2000	108	100MB	2

表 3: 3.5インチディスク装置の仕様

4. シミュレーション

4.1 シミュレーションの仮定

本稿ではディスクと主記憶は同じデータ構造を持つものとし、B木などは考えない。また更新結果をデータベースに反映するのに要するだけを時間を求め、データ演算等の時間は考慮しない。主記憶に更新結果を反映する場合は更新されたページを書き込む時間を、ディスクに反映する場合は、データの書き込み時間は無視できるほど小さいとして、ヘッドの移動に要する時間をそれぞれ計算によって求めた。以下の議論では、次の2つの方式のディスクデータベース、完全主記憶データベースおよび本稿の部分的主記憶データベースについて比較する。

Historyのレコードは確定した全てのトランザクションにより追加される。レコードの追加はディスクデータベースでは次節で述べる更新と同じように行なわれるとする。また、完全主記憶データベースではHistoryも主記憶に常駐させるものとし、部分的主記憶データベースではこのレコードに一時的にアクセスが集中するのでホットスポットデータと見なして主記憶に常駐させ、アクセスが集中しなくなったらHistoryディスクに書き込むも

のとする。本稿では、ディスクバッファや主記憶バッファはこの追加に対応できる程度の容量を持つとする。従って、ディスクアクセスにかかる時間を計算する場合は Account、Branch、Teller の更新に関してのみ考慮する。また、各ディスク装置はそれぞれ並列にアクセスできるものとする。

各ディスク装置当たりのアクセスは最適化されているとする。さらに、データのアクセスに 80/20 ルールが存在する場合、データの配置も最適化されているとする。すなわち、各ディスク装置に対して均等にアクセス要求が到着し、ホットスポットデータが存在するシリンダは連続している。

ディスク装置 1 台のの容量を $Size\_Disk$ とすると、Account、Branch、Teller を保存するディスク装置の数は $\lfloor C \cdot tps/Size\_Disk \rfloor$ となる。またシミュレーションの計測時間を T とすると、最大 $tps \cdot T$ 個のトランザクションが確定する (全てのトランザクションが確定する場合)。各トランザクションは 3 つのデータベースを更新するので更新回数の最大値は $3 \cdot tps \cdot T$ である。ディスク装置に対するアクセス分布は一様とするから、1 台当たりのアクセス数は以下の式で表される。

$$\frac{3 \cdot tps \cdot T}{\lfloor C \cdot tps/Size\_Disk \rfloor}$$

ここで議論を簡単にするために、$C \cdot tps/Size\_Disk$ が整数であるとすると、上式は

$$\frac{3 \cdot Size\_Disk \cdot T}{C}$$

となる。この式は、アクセスが各ディスク装置に一様に行われるような理想的な状況では、ディスク装置当たりのアクセス数は tps ではなく $Size\_Disk$ と T に比例することを示している。従って、$Size\_Disk$ はこのベンチマークのディスクアクセスに対応できるように小さくなければならない。3.2 節より、$Size\_Disk = 400$[MB]、$C = 10,001,100$ であるので 1 台当たりのアクセス数はおよそ $120 \cdot T$ となる。以下で、このアクセス数に対するそれぞれの方式の時間を求める。

4.2 シミュレーション結果

方式 1: 各トランザクションは確定するたびに更新結果をディスクに書き込む。

最初に、データベースのアクセスが一様である場合について考える。$Num\_Cylinder$ をディスク装置のシリンダ数とすると、平均シーク距離は $Num\_Cylinder/3$[TEO72] となるので、平均シーク時間は 17.9[ms] である。平均回転待ち時間は 60[min/s]/5000[rpm] $\cdot \frac{1}{2} = 6$[ms] となる。従って、計測時間 T の間に生成されたデータ更新をディスクに書き込むのに要する時間は次式で表される。

$$120 \cdot T \cdot (17.9[\text{ms}] + 6[\text{ms}]) \simeq 2.9 \cdot T[\text{s}]$$

上式はデータの書き込みに要する時間が計測時間の 2.9 倍もかかることを示す。

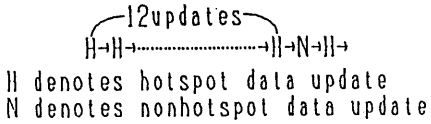

図 1: 更新要求の到着分布

次に、80/20 ルールが存在する場合について考える。$96 \cdot T(= 120 \cdot T \cdot 0.8)$ のアクセスがホットスポットデータの更新を要求し、$24 \cdot T(= 120 \cdot T \cdot 0.2)$ のアクセスが非ホットスポットデータの更新を要求する。また、データ更新は一様に到着するものとする。従って、図 1 に示すように、ホットスポットデータの更新が 12 回到着した後に非ホットスポットデータの更新が 1 回到着することになる。ホットスポットデータシリンダ間の平均シーク距離は 400/3 であるので、平均シーク時間は 10.7[ms] となる。ホットスポットデータと非ホットスポットデータ間の平均シーク時間は一様アクセスの場合と同じく 17.9[ms] とする。以上より、データ更新をディスクに反映するのに要する時間は次式で表される。

$$72 \cdot T \cdot (10.8[\mathrm{ms}] + 6[\mathrm{ms}]) + 24 \cdot T \cdot (17.9[\mathrm{ms}] + 6[\mathrm{ms}]) = 1.8 \cdot T[\mathrm{s}]$$

以上の 2 つの結果より、方式 1 では TPC ベンチマーク B が実行できないことが判る。

方式 2: 更新を主記憶に一時バッファリングし、複数のトランザクションの確定後にまとめてディスクに書き込む。

方式 2 はグループコミットに相当するものである。多くの更新が主記憶にバッファリングされるので同一のシリンダに対する書き込みが複数存在すると考えられる。従って、スケジューラによるディスクアクセスの最適化が行なえると期待できる。最初に、データベースアクセスが一様分布である場合について考える。このとき、あるシリンダがアクセスされる確率 p は $1/Num\_Cylinder = 1/2000$ となる。この確率はきわめて小さく、また、更新数 (Num_Update) はきわめて多いので、あるシリンダが x 回の書き込みを要求される確率 $P(x)$ はポアソン分布に従うと考えられる。ここで、$\lambda = Num\_Update \cdot p$ とすると、確率 $P(x)$ は次式で表される。

$$P(x) = \frac{\lambda^x \cdot e^{-\lambda}}{x!}$$

x 回の書き込みを要求されるシリンダがいくつ存在するかを示す期待値 $E(x)$ は次式で表される。

$$E(x) = Num\_Cylinder \cdot P(x)$$

上式を用いて計算した結果として、5 分間のバッファリングを行った場合の例を図 2 に示す。図 2 より 1 シリンダ当たりの更新数は多くても数百回程度であり、同一のセクタに対する更新が何度も要求される確率は比較的低いと考えられる。ここで、ディスクアクセ

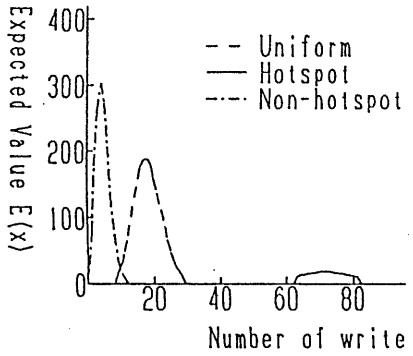

図 2: 一様分布および 80/20 ルールの条件でのアクセス分布

T[s]	5	10	15	20	25	30	35	40	45	50	55
Time[s]	6.66	12.3	17.3	22.0	26.3	30.2	33.8	37.0	40.0	42.6	45.0

T[min]	1	2	3	4	5	6	7	8	9	10
Time[s]	47.0	57.7	58.9	59.0	59.0	59.0	59.0	59.0	59.0	59.0

表 4: 一様アクセス分布におけるディスクアクセス時間

スは、初めに 1 つのシリンダに対して各ディスク面の順に行われ、次にシリンダの順に行われるものとする。j $(j \leq 4)$ 回の更新が 1 つのシリンダに反映される場合、ディスクの回転待ち回数は 0 から j の間で変化し、平均回転待ち回数は $j/2$ となる。また、$j > 4$ における平均回転待ち回数は簡単のため 2 とする。以上の議論より、ディスクアクセスに要する時間を計算した結果を表 4に、バッファリング時間 T に対する比率を図 3に示す。ここで、ディスクアクセスに書き込みが占める割合を 25%[SMI87] とすると、図 3において少なくともバッファリング時間に対する割合が 0.25 を下回るまでバッファリングをする必要がある。

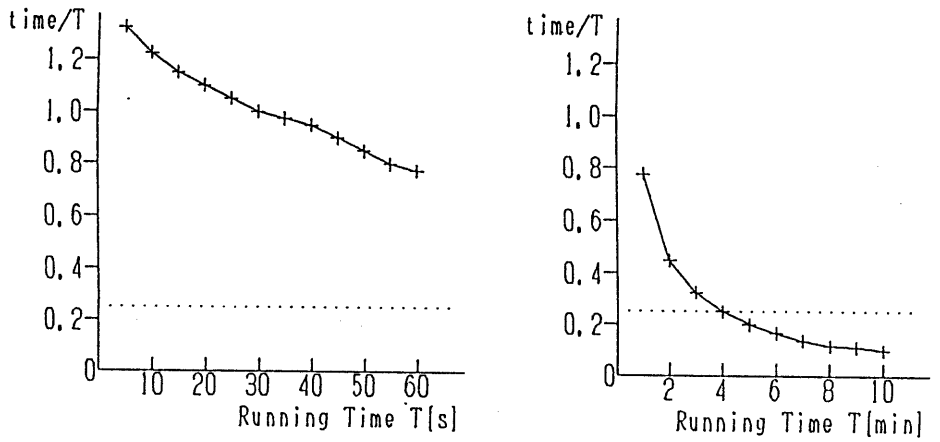

図 3: 一様アクセス分布におけるバッファリング時間に対する比率

T[s]	5	10	15	20	25	30	35	40	45	50	55
Time (H)[s]	4.39	7.41	9.41	10.6	11.2	11.5	11.7	11.8	11.8	11.8	11.8
Time (N)[s]	1.51	2.84	4.11	5.33	6.51	7.65	8.76	9.83	10.9	11.9	12.9
Total time[s]	8.90	10.3	13.5	15.9	17.7	19.2	20.6	21.6	22.7	23.7	24.7

T[min]	1	2	3	4	5	6	7	8	9	10
Time (H)[s]	11.8	11.8	11.8	11.8	11.8	11.8	11.8	11.8	11.8	11.8
Time (N)[s]	13.9	24.1	32.0	37.7	41.5	43.9	45.3	46.2	46.7	46.9
Total time[s]	25.7	35.9	43.8	49.5	53.3	55.7	57.1	58.0	58.5	58.7

H: ホットスポットデータ、N: 非ホットスポットデータ

表 5: 80/20 ルールにおけるディスクアクセス時間

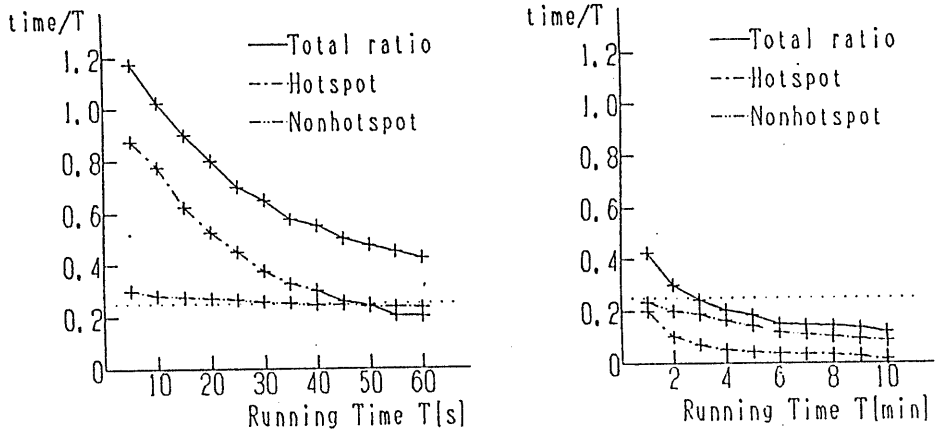

図 4: 80/20 ルールにおけるバッファリング時間に対する比率

次に、80/20 ルールの場合について計算した結果として、5 分間のバッファリングを行った場合のホットスポットデータと非ホットスポットデータの例を図 2 に示す。また、一様アクセスの場合と同様にディスクアクセスに要する時間の結果を表 5 に、バッファリング時間 T に対する比率を図 4 に示す。この結果、TPC ベンチマーク B の更新に対応するためにはある程度の時間のバッファリング (一様アクセスの場合 4 分以上、80/20 ルールの場合 3 分以上) を行えば良いことが判る。しかし、現実にはデータアクセスの最適化を行なうオーバヘッドも考慮しなければならない。

完全主記憶データベース

主記憶には最大 $4 \cdot tps \cdot T$ 回のデータ更新が行われる。また、主記憶に対する書き込みは並列化しないものとする。ここでページサイズを 1k バイト、バスサイズを 4 バイト、メモリのアクセスサイクルを 100[ns] とすると、更新結果をデータベースに反映するには、$1.0 \times 10^{-4} \cdot tps \cdot T (\simeq 4 \cdot tps \cdot T \cdot 1[\text{kbyte}] \cdot 100[\text{ns}]/4)$ の時間がかかる。これは書き込みアクセスの割合を 25% とし、データ処理などの時間を考慮しなければ、最大 2000tps 程度の処理ができることを意味する。

部分的主記憶データベース

これまでの議論より、主記憶には最大$4 \cdot tps \cdot T \cdot 0.8$回のデータ更新が行われる。ホットスポットデータの更新結果を主記憶に反映するには、$8 \times 10^{-5} \cdot tps \cdot T (\simeq 4 \cdot tps \cdot T \cdot 0.8 \cdot 1[\text{kbyte}] \cdot 100[\text{ns}]/4)$の時間がかかる。また、1ディスク装置当たり$24 \cdot T$回の更新がディスクI/Oを要求する。ここで、非ホットスポットデータの更新を方式2と同じように行なうとすると、、ディスクI/Oに要する時間は表5の非ホットスポットデータの場合と同じである。従って、書き込みアクセスの割合が25%であるなら、TPCベンチマークBを実行するには非ホットスポットデータの更新を少なくとも40秒間バッファリングしなければならない。しかし、実際には非ホットスポットデータの更新はそれほど頻繁に起こらないので、更新要求があるたびにディスクに反映できると考えられる。他にデータ処理時間等を考慮する必要があるため、実際には数分間のバッファリングを行なう必要があると考えられる。

このように、部分的主記憶データベースは非ホットスポットデータのディスクアクセスを最適化できれば、完全主記憶データベースと同程度のトランザクション処理能力を持つような高性能システムを実現できることを示している。また、ディスクデータベースとは異なり、非ホットスポットデータのみの最適化を行なえば良いので、そのオーバヘッドはかなり小さいと考えられる。

5. まとめ

本稿では、TPCベンチマークBを想定したシミュレーションを行ない、部分的主記憶データベースのトランザクション処理能力がどの程度まで期待できるかについて述べた。今後の課題としては、現在のシミュレーションでは処理にかかる時間を考慮していないので、この時間を含めた性能評価を行なう必要がある。

また、最近では、電源を切ってもデータを保持し続けるフラッシュメモリが注目されている。現在の製品は、ホットスポットデータの更新のように頻繁に書き変えを行なうことはできないが、将来このような書き変えにも対応できるフラッシュメモリが製品化されれば、ディスクをメモリに置き換えることも可能になると思われる。

参考文献

[AMM85] A.C. Ammann, M.B. Hanrahan, R. Krishanmurthy, "Design of a Memory Resident DBMS," IEEE Compcon Digest of Papers, 1985, pp.54-57.

[BER87] P.A.Bernstein, V.Hadzilacos, N.Goodman, "Concurrency Control and Recovery in Database Systems," Addison Wesley, 1987.

[DEW84] D. DeWitt, et al, "Implementation Techniques for Main Memory Database Systems," Proc. of ACM SIGMOD Conf., 1984, pp.1-8.

[EIC86] M.H. Eich, "Main Memory Database Recovery," ACM FJCC, 1986, pp.1226-1232.

[EIC87] M.H. Eich, "A Classification and Comparison of Main Memory Database Recovery Techniques," Proc. IEEE 3rd Conf. on Data Engineering, 1987, pp.332-339.

[EIC88] M.H. Eich, "MARS : The Design of a Main Memory Database Machine," Database Machines and Knowledge Base Machines, Kluwer Academic Publishers, 1988, pp.325-338.

[EIC89] M.H. Eich, "Main Memory Database Research Directions," Proc. 6th International Workshop, IWDM'89, 1989, pp.251-268.

[GAW85] D. Gawlick, "Processing Hot Spots' in High Performance Systems," Proc. of IEEE Spring Computer Conference, 1985, pp.249-251.

[GRA91] Jim Gray, "The Benchmark Handbook," Morgan Kaufmann Publishers, 1991, pp.19-117.

[HAG86] R.B.Hagmann, "A Crash Recovery Scheme for a Memory-Resident Database System," IEEE Trans. on Computers, Vol. C-35, No.9, September, 1986, pp.839-843.

[KAM91] Y. Kambayashi, H. Takakura, "Realization of Continuously Backed-up RAMs for High-Speed Database Recovery," The 2nd International Symposium on DASFAA, 1991.

[KIM85] W. Kim et al (Eds), "Query Processing in Database Systems," Springer-Verlag, 1985.

[LEH86a] T.J. Lehman, M.J. Carey "Query Processing in Main Memory Database Management Systems," Proc. of ACM SIGMOD Conf., 1986,pp.239-250.

[LEH86b] T.J. Lehman, M.J. Carey "A Study of Index Structures for Main Memory Database Management Systems," Proc. of 12th International Conf. on VLDB, 1986, pp.294-303.

[LEH89] T.J. Lehman, M.J. Carey "A Concurrency Control Algorithm for Memory-Resident Database Systems," Proc. 3rd International Conf, FODO, 1989, pp.490-504.

[SHA86] L.D. Shapiro, "Join Processing in Database Systems with Large Main Memories," ACM Trans. on Database Systems, Vol.11, No.3, 1986, pp.234-264.

[SMI87] A.J. Smith, " Sequentiality and Prefetching in Database Systems," ACM Transaction on Database Systems, Vol.3, No.3, 1987, pp.223-247.

[SON87] S.H.Son, "A Recovery Scheme for Database Systems with Large Main Memory," COMPSAC, Tokyo, Japan, 1987 pp.422-427.

[STR83] R.A. Scranton, D.A. Thompson, D.W. Hunter, "The Access Time Myth," IBM Technology Report, RC10197, September, 1983.

[TEO72] T.J. Teorey, T.B. Pinkerton, "A Comparative Analysis of Disk Scheduling Policies," Communications of ACM, 15:3, March, 1972.