

ウェアラブルコンピュータにおける 小型ディスクドライブの有効利用法の紹介

Introduction of Effective Use of Small Disk Drive for Wearable Computers

小野田 英樹

要 約 近年, PDA(Personal Digital Assistant)や携帯電話などのモバイルコンピュータが急速に普及しており, これらは将来的にはGPS(Global Positioning System)¹⁾やカメラを内蔵したウェアラブルコンピュータ²⁾として利用されると予想される. ウェアラブルコンピュータは, 利用者のアシスタントとして動作する事が望ましく, 機器が情報を能動的に利用者に提示する必要がある. また, 画像などをサイズの大きいデータを格納するストレージとして, 半導体メモリよりも低価格で大容量の小型ディスクドライブ(以下, 小型ディスクと呼ぶ)を利用することは適当である.

利用者に有益な情報を提示するには, 能動型データベースの利用が有効であるが, 状況の判断を常時に行う必要があるため, データの参照などのディスクアクセスが頻繁に発生する. 小型ディスクは, シーク時間, 回転速度が通常のディスクに比べ劣るため, データアクセス速度に問題がある. また, 近年, ウェアラブルコンピュータ上に能動型データベース管理システムを構築する研究が行われているが, それらは多様なウェアラブルコンピュータに対応できていない.

上記問題に対し, 小型ディスク内のディスク制御プロセッサとバッファメモリを利用し, データベースシステムを小型ディスク内で動作させることを提案する. これより, ソフトウェアレベルでの共通インタフェースを提供することができ, 多様なコンピュータで利用可能なシステムの構築が可能となる. さらに, 小型ディスク内のリソースをデータベース処理に利用することで, ウェアラブルコンピュータへの負荷を軽減することが可能となる. また, データアクセス時間を短縮するために, 小型ディスク上にデータベースのログを連続的に記録した追記型ファイルシステムを構築することを提案する. 提案する追記型ファイルシステムでは, シーケンシャルアクセスを利用することにより, データ処理能力が向上する事が考えられる. また, ログのスキャンにより, 過去から現在までの情報を検索するテンポラルクエリの実行, システム障害時におけるデータベース復旧が可能となる. 本手法の有効性を検証するため, 従来のページ単位アクセスによるファイルシステムとデータアクセス性能に関して比較実験を行い, 高速にデータアクセスが可能であることを確認した.

Abstract Recently, mobile computers, such as PDA(Personal Digital Assistants)and cellular phones are quickly widespread, and they will be used as wearable computers with a GPS(Global Positioning System) and/or a small camera. It is preferable that wearable computer operates as user's assistant. Therefore, the wearable computers have to actively present users appropriate information. To store image data, it is appropriate to use small disk drive as a storage device for wearable computers taking its capacity and price into consideration.

In order to provide valuable information to users, use of an active database is a good solution. But active database causes frequent disk accesses to judge the situation. In addition, data access speed of a small disk is low, because its seek time and rotational delay are inferior to those of desktop models. There have been some studies on building an active database management system on small computers, whereas their systems do not work with several kinds of small computers.

To solve the above mentioned problems, the author proposes the active database management system built on a small disk, in which a disk control processor and a buffer memory are used as a CPU and main memory, respectively. Our proposed system offers common interfaces to any small computers. This idea also reduces on the wearable computer resources, because the database management system does run on the small disk, not on the computer.

To reduce the data access time on a small disk, the author recommends all of database logs are recorded on the disk file sequentially (in write once mode) in a chronological order. The sequential access to the disk improves I/O throughput because it can reduce the head positioning time. The nature of the log in a chronological order enables temporal queries to be evaluated easily. Moreover, use of the write once (append only) file system brings another advantage that the recovery of database is quite simple to position the chronological logs. In order to evaluate and show the efficiency of the proposed write once file system, we have performed the experiments for comparing our proposed method with the conventional page based one.

1. はじめに

近年、小型ノートパソコンやPDA、携帯電話などの登場により、日常的にコンピュータを携帯することが可能となってきた。これらコンピュータを、利用者のアシスタントとして利用する事への要求が高まりつつある。そのため、利用者の私生活をさまざま側面から記録したライフログを利用することで、利用者が求めている情報を、状況に応じてコンピュータが判断し、能動的に情報を提示する必要がある。このように身に付けて利用するコンピュータは、ウェアラブルコンピュータと呼ばれている。ウェアラブルコンピュータが利用者をアシストする場面は、日常生活の場から災害救助現場など幅広く求められており、ネットワークが無いような環境においてウェアラブルコンピュータ単体で動作する必要性がでてきている。機器単体で動作するためには、必要なデータをすべて蓄えることが可能な大容量なストレージが必要となる。大容量データをウェアラブルコンピュータ内に蓄えるストレージとして、低価格の小型ディスクを利用することは適当である。

利用者が必要としている情報を能動的に提示するには、あらかじめ登録したルールに基づき動作し、ルールの追加、削除により、提示する情報のカスタマイズが可能な能動型データベースの利用が有効である事が知られている。しかし、能動型データベースは通常のデータベースに比べ、イベント発生時のイベント処理、コンディションの評価および評価に基づくアクションが、システム側で自動的に実行されるため、データの参照、イベントの監視など、ディスクアクセスの頻度が多くなり処理コストが高くなる。しかし、小型ディスクはサーバ用ハードディスクドライブと比べシーク時間、回転速度が劣るため、データアクセス速度は低い。そのため、迅速な情報の提供には小型ディスクの特性に見合ったデータベースシステムの設計が必要となる。

本稿では、小型ディスクにおけるアクセス速度の問題に対して、データベース操作によって

作成されたデータを小型ディスク上に連続的に記録する手法（以下，追記型ファイルシステムと呼ぶ）を採用し解決を図る．追記型ファイルシステムでは，データベース操作によるすべてのデータがディスク上に保持されているため，ディスクを読み直すことでデータベースをリカバリすることが可能となる．また，連続的にデータが格納されるため，ディスクヘッドの移動回数を減らすことにより最適な性能を引き出すことが可能となる．さらに，個人で利用するシステムでは，時間をパラメータとする問合せ要求が考えられる．提案するシステムでは，全データをディスク上に時系列で保存しているため，過去の情報を振り返って検索するテンポラルクエリを効率的に処理することが可能となる．

提案する追記型ファイルシステムを実装するため，データ格納方法，クエリ処理方法，インデックスの作成方法について具体的に示し，ディスクアクセス性能について検証を行う．

2. ウェアラブルコンピュータ，小型ディスクと能動型データベース

ウェアラブルコンピュータは，利用者が身につけて利用するため，小型化，軽量化そして消費電力を低減する必要がある．そのため，通常のコンピュータに比べ処理能力が著しく劣る問題がある．また，ウェアラブルコンピュータが利用者に情報を提示するためには能動型データベースの利用が有効である．しかし，能動型データベースはデータアクセスを頻繁に行うため，小型ディスクが効率よくデータのアクセスを行う必要がある．これら問題点の詳細について述べる．

2.1 ウェアラブルコンピュータ

ユビキタス・コンピューティング^[31]の一部にウェアラブルコンピュータが位置づけられている．ウェアラブルコンピュータは，文字通り身につけて利用するコンピュータのことを指しており，マサチューセッツ工科大学 (MIT) のメディアラボで提唱された概念である．

ウェアラブルコンピュータは利用者が常に身につけて使用することを想定しているため，機器の装着による利用者の負担を軽減するため，小型化，軽量化されている．また，電源のない場所での長時間の使用にも耐えうるため，消費電力を抑えることも必要となっている．そのため，ウェアラブルコンピュータ内に高性能 CPU，大容量メモリや大容量ストレージを収めることができず，設置型コンピュータに比べ処理能力，記憶容量に問題がある．

ウェアラブルコンピュータにおいて，記憶容量が少ない問題を補うために，機器に挿入して利用する小型ストレージが利用されている．たとえば，コンパクトフラッシュ，小型ディスクやメモリカードがあげられる．

2.2 小型ディスク

小型ディスクは従来のハードディスクドライブと構造的には同様であるが，コンパクトフラッシュサイズとなっている．ディスクは多層化すると同時に記録することができるディスク枚数が増加するため，データ読み書き速度が速くなる．しかし，小型ディスクは，コンパクトフラッシュサイズに収めるため，ディスクは多層化されていない．このため，小型ディスクは，サーバ用ディスクドライブに比べデータアクセス性能において劣っている．日立のディスクドライブ^{[41][51][6]}における現状の性能を表 1 に示す．

表 1 から読み取れる問題の一つに，ディスク回転数があげられる．一般に，回転体は高速に

回転するほど振動に弱くなる傾向がある。また、回転数が上がるほど消費電力が増加する。小型ディスクは耐衝撃性や消費電力などの問題から高速なディスク回転速度が得られないため、ディスクアクセスが低速である。実際ディスク容量は増加しているが、ディスク回転速度の向上は見られない。回転速度が上げられないことにより、ディスクヘッドが目的とするレコード位置に移動するアクセス時間は約 20.33 ms(平均シークタイム(12 ms)+平均待ち時間(8.33 ms))とシステムからみると多大な時間を要する。

一方、ライフログを、MPEG 4^[7]の規格である 384 kbps のビットレートで記録した場合、カタログ上の性能ではディスクへの転送速度に問題はなく、24 時間記録した場合、約 4 GB の容量で収めることができる。MPEG 4 はオープンな規格であるため、ウェアラブルコンピュータの分野で採用されつつある。したがって、小型ディスクは性能に着目すれば、ウェアラブルコンピュータ上で有効に利用することが可能である。

表 1 ディスクドライブの性能 [日立 HP より抜粋]

	小型ディスク DSCM-11000	小型ディスク HMS360404D5CF00	サーバーモデル Ultrastar 5K147-147
容量 (GB)	1	4	147
密度 (Gb/inch)	15.3	56.5	---
転送レート (Mb/s)	36-60	57.1-97.9	1129
連続読み書き速度 (MB/s)	2.5-4.1	4.3-7.2	---
平均シークタイム (ms)	12	12	3.7
平均待ち時間 (ms)	8.33	8.33	2.0
回転速度 (rpm)	3600	3600	15000
データバッファサイズ (KB)	128	128	16384
メディア転送速度 (Mb/s)	38.8-59.9	57.1-97.9	---
インタフェース転送速度 (Mb/s)	11.1	33	2560
消費電力 (mW)	420	360	11200

2.3 データベース

2.3.1 データベース管理システム

データベースを利用する際、特に重要な項目が耐障害性である。データベース障害によりデータが消失すると、データを利用できないばかりか、正しくない結果を提示してしまう可能性がある。したがって、データを消失させない仕組みの実装が重要となる。通常、データベースはログを利用してその問題を解決している。ログとは、データベース操作によりデータベースに変更が生じた際、記録する変更データのことである。

データを消失させないため、データベース管理システムは、作成したログを永続メモリに記録したのち持続メモリの更新を行う。永続メモリ上のログは条件が満たされるとディスクにフラッシュされる。システム障害が発生した場合、永続メモリ上にデータが保存されているためデータベースを復元することが可能となる。

2.3.2 能動型データベース

能動型データベース^[8]は、ルールベースとデータベースをあわせもち、あらかじめ指定された処理を自動的に行う機構をデータベース内に持つ自律性を持ったシステムのことを指す。能動型データベースに対する処理を記述するルールとして、ECA(Event Condition Action)ルールが用いられる。

Event

トリガされるルールの原因となるイベントを記述する．

例：SQL 文の検索，挿入，更新，削除などのデータベース操作

Condition

ルールがトリガされる度に，参照される条件を記述する．

例：SQL 文の Where 節にあたるデータの制約条件

Action

トリガされたルールのうち，条件参照結果が真の場合に実行されるイベントを記述する．

例：SQL 文の検索，挿入，更新，削除などのデータベース操作，Abort，Commit などのトランザクション操作

能動型データベースは，監視対象であるイベントの発生時，ルールをトリガしコンディションが真であればアクションを実行する．このため，データベース管理システムは，イベントの状態遷移を記憶しておく必要がある．イベントの状態遷移は，正確な情報を提示するため消失してはならない．また，能動型データベースは，イベントが発生するたびにシステム側でデータベース処理を実行するため，一般のデータベース管理システムより処理コストが高い．ウェアラブルコンピュータで，能動型データベースを利用するには，データの格納する小型ディスクのアクセス性能の改善が必要となる．

3. ウェアラブルコンピュータのためのファイルシステムの提案

本章では，小型ディスクの特徴について検討し，ウェアラブルコンピュータ上で小型ディスクを利用する際の問題点の解決方法について提案を行う．ウェアラブルコンピュータは，その利用方法による制約から，十分な処理能力が得られない．従って，小型ディスク内のリソースを有効活用することによって，ウェアラブルコンピュータの内部リソースに処理を頼ることなく，小型ディスク内で処理を行うことが可能となる．小型ディスク内で処理を行うことにより，共通のソフトウェアレベルでのインタフェースを小型ディスク内に設けることが可能となり，ディスクに自律性を持たせることができる．概要を図1に示す．さらに，個人で利用するウェアラブルコンピュータで，小型ディスクを使用した場合に，高速にアクセスが可能となるファイルシステムを構築する方法について述べる．

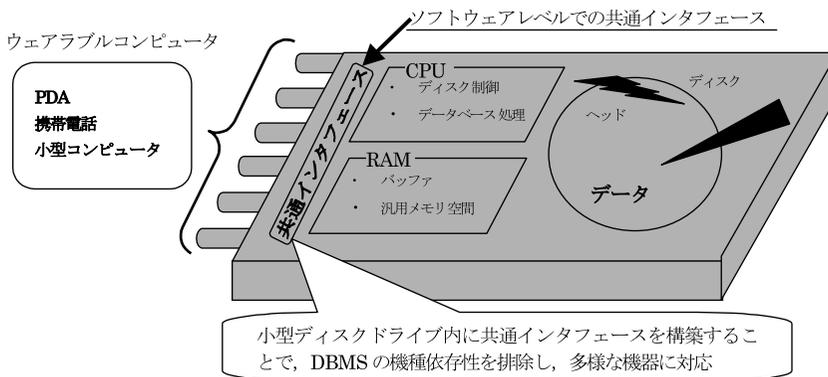


図1 小型ディスク上のシステム構成

3.1 記録方式

小型ディスクのアクセス時間を短縮するために、追記型ファイルシステムを利用する。提案する追記型ファイルシステムは、データの処理が発生し、記録するデータをディスクに連続して追記していく手法をとる。このため、ディスク上にデータを隙間なく記録することで効率よくディスクを利用することができ、データ追加時には、従来のデータベース管理システムで発生するディスクのランダムアクセスは起こらない。そのため、計算上、小型ディスクのアクセス時間内(20.33 ms)に約115 KBのデータを転送することが可能となり、効率の良いデータアクセスが実現可能となる。データベースは、2.3.1項で述べたように、ログが保存されていればリカバリが可能である。提案する追記型ファイルシステムでは、データ更新する際、更新前データの削除は行わず、更新データをディスクの追記部分に記録する手法をとる。また、データを削除する場合、実際にはデータをディスク上から消さず、データの格納位置を示すインデックスの削除により、データを削除したかのように見せかける手法を採用する。したがって、データ操作による全データを記録しているため、追記型ファイルシステムはリカバリの観点からも有効な手法である。

また、サイズの小さいデータを頻繁にディスクに書き込む場合、ディスクアクセスのオーバーヘッドが高くなるため、シーケンシャルアクセスの利点が十分に得られない可能性がある。このため、まとまったデータを一括して書き込むグループコミット^[8]の手法を採用し、効率的なデータ処理を実現する。メモリ上にログバッファを設け、データの変更情報は一旦ログバッファに書き込む。ログバッファがデータで満たされると、ディスクにデータを書き込む。ウェアラブルコンピュータで扱うデータには、位置情報など、たとえデータが消失しても前後のレコードから推測することが可能なデータがある。これらのデータは重要度が低いいため、データの消失が発生しても問題がないと考えられる。一方、所持金情報など、重要度の高いデータをメモリ上に蓄えると、障害によりデータが消失する危険性があるため、ディスクに強制的にフラッシュする。このようにデータにプライオリティ(重要度)を持たせることで、ディスクフラッシュの制御を行う必要がある。

3.2 バッファメモリの利用

小型ディスク上のバッファを汎用メモリとして利用する。メモリ上にはディスクに格納したデータのデータ格納位置を示すインデックスを展開する。一般にインデックスはディスク上のデータを検索する際、高速に処理を行うために利用する。しかし、インデックスは頻繁に更新されるため、ディスク上にインデックスを作成するとインデックスの読み込み、更新、変更の際、ディスクアクセスが必要となり、ランダムアクセスが発生する。このため、シーケンシャルアクセスの利点を十分に利用することができない。提案するファイルシステムでは、インデックスは小型ドライブ上のメモリに展開することで、この問題に対処する。メモリ上にインデックスを保持した場合、電源障害などによりインデックスの消失が考えられる。提案する追記型ファイルシステムでは、データの操作履歴がディスク上にすべて記録されているため、インデックスの復旧は容易である。

ウェアラブルコンピュータに小型ディスクを装着して利用した場合を想定する。現在、GPSにより計測できる精度は水平方向で10 cmになっている。実際に外出先で機器を利用した場合、精度は10 m程度で問題ないと考えられる。したがって、歩行時で約10秒に一度のデー

データの追加が行われると利用価値あるデータとなると考えられる。また、外出時に利用するデータテーブルは、例えば、位置情報テーブル、店舗情報テーブル、利用者の動作を記録する個人プロフィールテーブルなどが挙げられる。上記3テーブルに10秒おきに操作がされると仮定すると、約26,000レコードが24時間で蓄えられ、このレコードを示すインデクスのサイズは105KBとなり、小型ディスク内のメモリに収めることが可能である。自宅から持ち出す初期レコードは、利用者の個人データであると考えられるため、全レコード数が著しく増加することは考えにくく、全インデクスをメモリ上に構築することは十分可能である。

3.3 データ格納方式

データ格納方式は2種類考えられ、インデクスの作成方法がそれぞれ異なる。処理形態をそれぞれ図2図3、に示す。インデクスはディスクに格納されたレコード位置をキーとしている。図2は、データ更新時、レコード全体を複製し追記する方式で、インデクスのポインタは追記されたレコードを示す。これを複製方式と呼ぶ。一方、図3は、データ更新時に更新された属性情報のみを追記する方式で、もともとのレコードへのポインタは保持したまま、更新された属性へのポインタを新たに作成し元レコードへのポインタにリストとして追加する。元データへのポインタと変更データの格納先へのポインタにより最新レコードの作成が可能である。これを差分方式と呼ぶ。

3.3.1 複製方式

データの変更が発生した場合、変更の対象となるレコードの未変更データも含む全体を複製し、ディスクに追記する方式である。

1) 挿入

新たなレコードを挿入する際、新たに一意のレコードIDを発行する。レコードIDと、レコード長、データをひとつの連続データとしてディスクに追記する。データ格納後、格納位置へメモリからインデクスを張る。このとき、追記したレコードの先頭アドレスをキーとしてインデクスを記録する。従来のデータベース管理システムでは、データを挿入する際、格納可能な空き領域の検索を必要とする場合があるため、ランダムアクセスが発生し、データ格納に時間を要することがある。提案手法では、データを格納する追記場所があらかじめ決まっているため、ランダムアクセスは発生しない。そのため、効率よくデータを追加することが可能である。

2) 削除

追記的にレコードを記述しているため、削除するレコードIDのみをディスクに追記する。その後、メモリ上のインデクスを削除する。提案する追記型ファイルシステムでは、実際にデータをディスク上から削除することはなく、削除情報のみをディスクに追記する。データ問合せ時には、インデクスを利用してレコード位置を求めるため、インデクス情報を削除することでシステム側からはデータが削除されたと判断することが可能である。従来のシステムは削除の実行時、ディスク上からデータの削除を行うが、提案システムでは削除処理も追記データしてディスクに追記する。

3) 更新

メモリ上のインデクスを利用して対象レコードを特定し、レコードIDを取得する。取

得したレコード ID, 更新情報を含むよう複製したレコードをディスクに追記する. この際, 更新前データをディスク上に残したまま, 更新前データを示しているインデクスを追記されたレコード位置を示すように変更する. ディスク上には, 更新前データと更新後データが存在することになる.

4) 問合せ

インデクス情報に最新のレコードの格納位置が記録されているため, ディスク上のレコード記録場所を前もって特定することができる. ディスクアクセスを行う前に, インデクスを用いて必要なレコードの格納場所を収集し, レコード位置情報をディスク格納アドレス順にソートを行う. ソートを行うことでヘッドの戻りがないアクセスが可能となる. シーケンシャルアクセスでは, アクセス時間内(20.33 ms)に約 115 KB のデータを転送することが可能であるため, 問合せに不要な領域(必要レコードから次の必要レコードまでの領域)が, 115 KB 以下であれば, シーケンシャルアクセスによる読み飛ばしを行い, 領域が 115 KB 以上であれば, ヘッドを目的レコードへ移動させるランダムアクセスを行う.

複製方式では, 処理が単純なうえ, インデクスはレコード数以上に増えないためメモリの節約が行える. しかし, レコード全体を複製して記録するため, ディスク使用効率は悪い. 小型ディスクの性質上, ディスク領域より, メモリ領域の資源のほうが少ないため, 有効な手法ということができる.

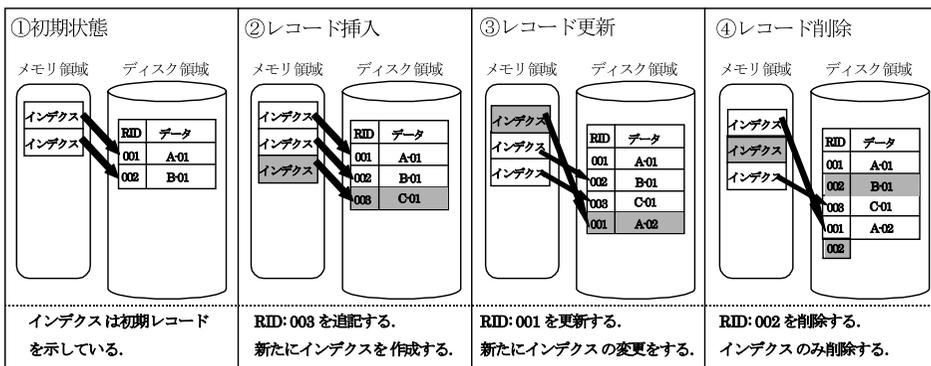


図2 複製方式のデータ格納方式とインデクス

3.3.2 差分方式

データ更新時には, 差分だけ追記する方式で, もととのレコードへのポインタは保持したまま, 更新された属性をディスクに追記し, 書き込まれた属性へのインデクスを作成する. 作成されたインデクスはレコード全体を示すインデクスに付け加える方式とする.

差分方式では, ディスクに記録するデータは更新された属性データのみなので, ディスクの節約が可能である. しかし, 属性の変更のたびにインデクスが作成されるため, メモリ使用において効率が悪い. メモリはディスクの比べ, 記録単位あたりのコストが高いため, 差分方式は複製方式で小型ディスク内に収められないデータ量を取り扱う時に利用するのが望ましい.

4. 基本実験と追記型ファイルシステムの検証

3章で提案したファイルシステムを実装するにあたり, 小型ディスクのアクセス性能の検証

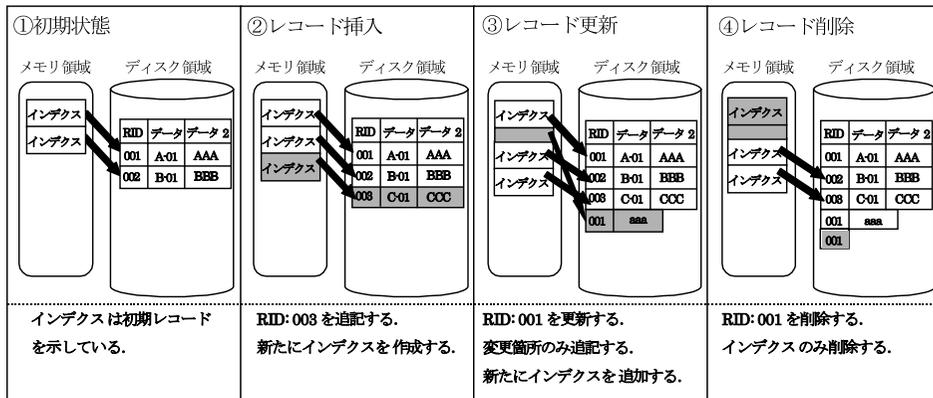


図3 差分方式のデータ格納方式とインデクス

を行った。また、インデクスを利用したデータ複製方式の追記型ファイルシステムを構築し、検索処理を行うことで、アクセス性能の検証を行った。現在のところ、内部リソースがユーザに公開された小型ディスクは存在しないため、Linux 上でシステム構築、およびテストのシミュレーションを行った。なお、Linux として、gentoo 1.4.1(kernel Version 2.6)、PC として IBM ThinkPad(CPU : pentium 3, Memory : 384 KB)、小型ディスクとして、表 1 に示すマイクロドライブ(1 GB)を使用した。提案するデータベース管理システムの処理は PC 内部の CPU、メモリを使用して処理を行い、データは提案手法に基づいて小型ディスクに格納した。

提案する追記型ファイルシステムを構築するため、C 言語の低水準入出力を利用して raw device としてディスクにアクセスを行う。低水準入出力ではディスクをシーケンシャルにアクセスすることが可能であり、OS があらかじめ備える。バッファリング機能を利用せず、直接ディスクとデータ転送を行うことが可能となる。一方、高水準入出力を利用すると、OS のバッファリング機能を利用し、ディスクの読み込み場所からバッファサイズに余裕がある限りデータの蓄積を行う。これによりバッファ内のレコードを利用する際、ディスクアクセスが必要なく高速に処理を行うことができる利点があるが、バッファに蓄えるためのデータを読み込むため、ディスクヘッドがバッファ容量分だけ移動する。OS のバッファリング機能を利用すると、ヘッドがバッファ容量分移動しているため、最適なディスクアクセスを行うことができない。提案する追記型ファイルシステムでは、ヘッド移動回数を抑えるため、3.3.1 項の 4) で述べたとおり、問合せに不要な領域(読み飛ばし領域)が 115 KB を境に、シーケンシャルアクセスとランダムアクセスを使い分けることにより、最適なディスクアクセスを実現する。また、raw device としてディスクを扱うため、ディスクヘッドは最終アクセスを行ったディスクトラック上にあり、レコード追記場所へのアクセスを平均回転待ち時間(8.33 ms)で行うことが可能となる。

4.1 使用データ

実験では、ウイスコンシンベンチマークで利用されるレコードセットを用いてディスクアクセス性能について検証する。ウイスコンシンベンチマークは、208 バイトのデータを 1 レコードとしてデータの読み書きを行うテスト方式である。1 レコードは、整数型変数(4 バイト)13 個と 52 文字からなる文字型配列(52 バイト)3 個から構成されている。レコードサイズが 208

バイトの場合、日立のマイクロドライブ(表1)のカタログスペックから計算を行うと、アクセス時間(20.33 ms)内にシーケンシャルアクセスでは約450~700レコードの読み込みあるいは書き込み処理を実行することが可能である。

4.2 基本実験

4.2.1 ディスクの基本性能

使用する小型ディスクの基本性能を調べるため、ディスク全体にデータを書き込む処理を行う。この処理結果から、以降の実験のパラメータを決定する。次に、ページ単位とレコード単位でのディスクアクセスの処理時間の比較を行う。レコードとは任意のサイズの単一データを示し、ページとは特定サイズのデータ集合で、一般的に数レコードのデータが含まれる。

頻繁にディスクアクセスを繰り返すと、ディスクアクセスのオーバーヘッドが生じる。このため、ページ単位アクセスを利用し、一括してデータを書き込むことで、ディスクアクセス頻度を減らすことができ、効率よくデータ処理を行うことが可能となる。なお、ページ単位のサイズは、約9レコードを一括して書き込む2KBのページサイズ、19レコードを一括して書き込む4KBのページサイズ約39レコードを一括して書き込む8KBのページサイズでアクセス速度を比較する。

はじめに、ディスクの基本性能の検証を行った。10MB単位のレコードを読み書きすることで計測を行った。結果を図4に示す。書き込みと読み込み双方でほぼ同等の性能が得られた。データ処理初期段階では小型ディスク内のバッファが利用されるため、高速に処理が行われていることが結果から判断できる。ディスク内バッファを利用しない状態で、平均して約6秒で10MBのデータを読み書きしているため、約1.7MB/secの読み書き性能となる。表1で示されたカタログ上の連続読み書き速度(2.5~4.1MB/sec)と比較して、40%~70%の性能が出ていることが判明した。1.7MB/secでの記録が行えるため、映像などの蓄積を行うウェアラブルコンピュータで、MPEG4の規格である384kbpsのビットレートで記録を行うことが十分可能である。また、映像を記録させながら、問合せを行うことも可能であると考えられる。

次に、ページ単位処理におけるデータサイズ変更による応答時間の計測を行った。4KBをページサイズとし、バッファを用いたディスクアクセスと、バッファを利用しないディスクアクセスの比較結果を図5に示す。4,000レコードを書き込む際、ページ単位アクセスでは68.9ms、レコード単位アクセスでは227.7msの時間を要している。したがって、バッファを用いる方がディスクアクセスのオーバーヘッドを大幅に抑えることが可能となる。

次に、バッファサイズを2KB、4KB、8KBに変化させ、ディスクアクセス時間の比較を行った。結果を図6に示す。2KBのバッファを用いた場合、ディスクアクセスのオーバーヘッドが発生していると思われる応答時間が得られた。バッファサイズを4KBと8KBに変化させた場合、応答時間に違いは見られなかった。したがってメモリ上に蓄えるデータ量が少ない4KBのバッファを用いる事が適当と考えられる。

4.2.2 追加と検索処理

以上の基礎実験から、バッファサイズを4KBとしたページ単位のシーケンシャルアクセスが最も効率の良い処理方法ということが判明した。ウェアラブルコンピュータでは、データ検索に比べ、データ追加処理が頻繁に発生するため、ヘッドはディスクの追記開始トラック上に

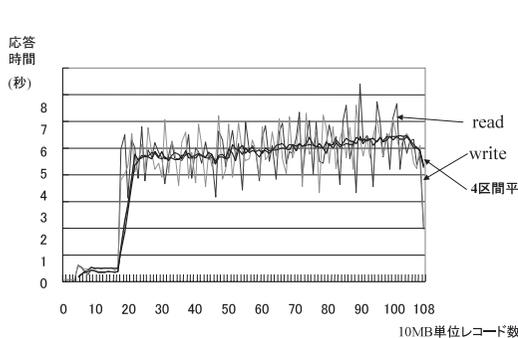


図4 10 MB 単位でのデータの読み書き時間

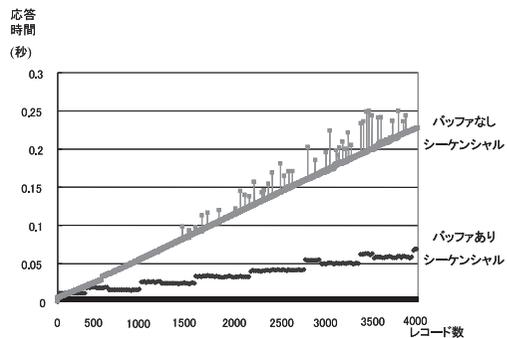


図5 レコード単位とページ単位の書き込み

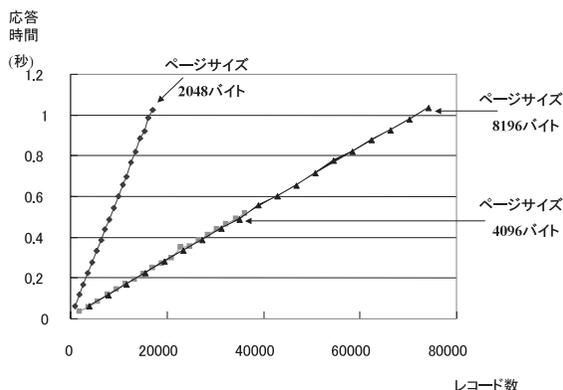


図6 バッファサイズを変化させた場合の応答時間

ある可能性が高いと考えられる。このため、追加処理を実行する場合、平均回転待ち時間で書き込みを開始することが可能である。ヘッド移動が最小限のため、追記処理時間は最大でもランダムアクセスによる処理時間を上回ることはない。

一方、データの検索時は、ディスクサーチが必要となる。データベースシステムでは検索性能を上げるため、一般にインデクスを用いる。提案システムにおいてもインデクスを利用し高速にディスクアクセスを行うことを提案している。アクセス対象レコードがディスク上に散在し、不必要領域が大きくなった場合、シーケンシャルアクセスによる不要データ領域を読み飛ばす時間が、ランダムアクセスによるヘッド移動時間を上回る事が考えられる。その場合、ディスクヘッドの移動によるランダムアクセスが有効となる。そのため、3.2節で提案したインデクスを有効に用い、最適な検索方法をとる必要がある。

4.3 実験内容

インデクスはディスクに格納されたレコード位置をキーとして張られているため、レコード位置をあらかじめ特定することができる。各レコードの格納場所を、アドレス順にソートすることでディスクアクセスを行う。アクセス対象レコードをアドレス順に検索するため、ヘッドの移動を最小限に抑えて検索をおこなう事が可能である。また、必要とするレコード間隔はインデクスをもとに求めることができ、レコードの間隔が広く、シーケンシャルアクセスによる読み飛ばしに要する時間がランダムアクセスによるヘッド移動時間を上回る場合、ランダムア

クセスを採用する。カタログ上の性能では、アクセス時間は 20.33 ms であるが、基礎実験から、本実験でのアクセス時間は 19.84 ms ということが判明している。19.84 ms 内にシーケンシャルアクセスにより読み取ることのできるデータ量は約 112 KB で、208 バイトのレコードでは約 600 レコードにあたる。すなわち読み飛ばし領域が 112 KB 以上か 600 レコード以上の場合、ランダムアクセスを利用することが有効である。

全レコード数を 1,000,000 レコードとして 300 回の検索処理を行い応答時間の測定を行った。必要なレコードの位置を、1 番目のレコード格納アドレスは 1, 2 番目のレコード格納アドレスは 4, 3 番目のレコード格納アドレスは 9, n 番目のレコード格納アドレスは n^2 となるよう配置し検索を行った。つまり、ディスクの検索の初期段階では、必要レコードが密に存在することになり、シーケンシャルアクセスによる読み飛ばしが利用され、必要レコードが疎になる後半部ではヘッドの移動によるランダムアクセスが行われるよう設定をおこなった。図 7 に示すように、すべてシーケンシャルアクセスは、初期段階ではレコード間隔が狭いため、ランダムアクセスに比べアクセス時間が早くなる。レコード間隔は徐々に広がるため、応答時間が二次関数的に増加する。すべてランダムアクセスを行った場合、ヘッドの移動が発生するため、アクセス時間が掛かるが、不要レコード間隔が広がるとシーケンシャルアクセスに比べ、ランダムアクセスによるディスクヘッドを移動させたほうが目的レコードに到達する時間が早くなる。その分岐点は約 230 レコードとなりレコード間隔は約 460 レコードである。ページ単位アクセスでは約 24 ページ目となり、24 ページ目からランダムアクセスを利用する方が良いことがグラフから読み取れる。レコード数をさらに増やした場合、24 ページ目以降はランダムアクセスのみの検索となる。ランダムアクセスの場合も、レコード間隔が広がっていくとディスクヘッドを磁気ディスクの目的の場所まで動かす時間は増加するが、シーケンシャルアクセスによる読み飛ばしによる時間の増加に比べ、十分小さいため、有効な手法である。

以上の実験から、ランダムアクセスとシーケンシャルアクセスを使い分けることで常に最適なディスクアクセスを実現することが可能となる。ウェアラブルコンピュータは、個人利用のシステムであるため、読み飛ばし領域の広い検索処理が発生する機会は少ないと考えられるため、提案システムは有効であると考えられる。また、レコードサイズが小さい場合、シーケンシャルアクセスによる読み飛ばすレコード数が増加するため、より高速にデータベース処理を行うことが可能となる。

また、ウェアラブルコンピュータで小型ディスクを利用するにあたり、消費電力の問題を考慮する必要がある。ディスクドライブが電力を消費する動作として、ヘッドの移動、ディスクの起動が上げられる。提案システムでは、ディスクをスケジューリングしてからアクセスするため、ヘッドの移動は最小限に抑えることが可能である。また、小型ディスクはディスクアクセスが行われない時、消費電力を抑えるためディスクの回転を停止する。ウェアラブルコンピュータで小型ディスクを利用した場合、数秒おきに GPS データなどが追加されるため、ディスクの停止と起動を繰り返し、ディスクにトルクを与えるための起動電力を抑えることができる。したがって、消費電力の面においても、提案システムは有効である。

5. 関連研究での本提案の位置づけ

以下に提案システムの特徴を確認するために関連研究を列挙する。

スマートカード上でのデータベース

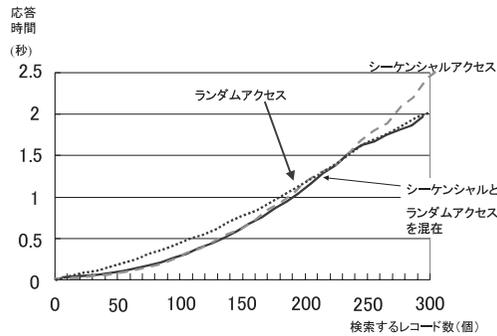


図7 検索処理比較

Bobineau らは PicoDBMS¹⁰⁾ のなかで 1 MB を超える非接触型スマートカード上で効率よく問合せを処理する RDBMS について論じている．スマートメディア上でのデータベースは、既存のリレーショナルデータベースやファイルシステムの拡張であるといえる．

PicoDB はスマートカードのみを対象としているが、我々のシステムでは、ディスクを対象としている．また、データを連続的に書き込むため、コンパクトフラッシュのひとつのデータブロックにアクセスが集中するといった問題を回避できることから、コンパクトフラッシュにも応用が可能である．

倉光らは、非接触型スマートカード上に小型の DBMS を設計、構築する方法¹¹⁾ について論じている．非接触型スマートカードは記憶容量が小さいため、認証されたホストとカードの上で分散的に DBMS を構築する手法を提案している．利用者は非接触型カード上のストレージで最低限のデータを持ち歩く．

提案システムでは、十分にストレージ容量が大きいため、データベース管理システム全体とデータをストレージ内に収めることができ、小型ディスク内でクローズしたシステムとなる．

自律ディスク

横田らはディスク装置の制御用プロセッサやキャッシュメモリを利用することでストレージ機器内で負荷分散、障害対策、障害回復を実現する手法¹²⁾ を提案している．

提案システムでは、ストレージ上のリソースを利用するという点は類似しているが、制御用プロセッサをディスククラスタの管理に利用するのではなく、データベースの処理に利用する点で異なっている．

Log Structured File system

Rosenblum は、ファイルシステムとしてログ型の記憶モデル¹³⁾ を利用することを提案している．ファイル削除が実行されると、ストレージに断片的に未使用空間が作成される．ストレージを効率的に利用するため、随時ガベージコレクションを実施する．

提案システムでは、過去から現在までの情報を検索するテンポラルクエリの実行を想定しているため、データの削除を行わない．これにより未使用空間が分散的に発生することは無く、ガベージコレクションは行わない．

6. おわりに

今後、ウェアラブルコンピュータがますます利用されることが予想される。ウェアラブルコンピュータは多様化しており、ウェアラブルコンピュータに依存したシステムの開発がされている。また、ウェアラブルコンピュータの小型化、軽量化に伴いウェアラブルコンピュータで利用されるストレージの小型化が進んでいる。小型ストレージの中に小型ディスクがある。小型ディスクは、容量と価格からウェアラブルコンピュータでの利用に適している。小型ディスクは高容量化されているが、ディスク回転速度面では変化が見られない。また、小型ディスクは持ち運びを想定しているため、耐衝撃性や消費電力の問題から十分に回転速度を上げられないことが原因となっている。

本稿では、小型ディスク上にデータベースを収めることにより、データベースが動作ウェアラブルコンピュータに依存するといった問題の解決法を提案した。これにより、ウェアラブルコンピュータにおいて、ディスクの差し替えだけであるウェアラブルコンピュータから他のウェアラブルコンピュータへと移動可能なデータベースシステムを利用することが可能となる。脆弱なウェアラブルコンピュータの処理能力を補うために小型ディスク内リソースの利用を提案した。データベース処理を小型ディスク内で行うことによりウェアラブルコンピュータのリソースの使用を抑えることが可能となる。また、小型ディスク上に連続的にデータを記録する追記型ファイルシステムを提案することで、データベースシステムから高速にアクセスが可能なファイルシステムの構築を提案した。ディスクアクセス時間について実験を行うことで、提案システムの実用性を示した。

謝辞

本稿は、筆者が奈良先端科学技術大学院大学情報科学研究科在学中に研究したものであり、植村教授をはじめ、宮崎助教授、波多野助手、天笠助手、データベース学研究室の方々にも多大なる御指導、御提案、御助言を賜ったことによる。ここに記して謝意を表わす。

-
- 参考文献**
- [1] SDGPS SkyFix XP
http://www.thales_geosolutions.com/Thales%20geo/skyfixxp/skyfixxp%20home.cfm
 - [2] Wearable
<http://www.media.mit.edu/wearables/index.html>
 - [3] Ubiquitous
<http://www.Ubiq.com/weiser>
 - [4] HITACHI
 HITACHI マイクロドライブ製品仕様 DSCM シリーズ
<http://www.hgst.com/japanese/products/microdrive/files/dscm.pdf>
 - [5] HITACHI
 HITACHI マイクロドライブ製品仕様 3 K 4 シリーズ
http://www.hgst.com/japanese/products/microdrive/files/3_k_4.pdf
 - [6] HITACHI
 HITACHI サーバ向ハードディスクドライブ製品仕様 Ultrastar 15 K 147 シリーズ
http://www.hitachigst.com/portal/binary/com.epicentric.contentmanagement.servlet.ContentDeliveryServlet/JP_Public/products/Ultrastar/15_K_147/15_K_147.pdf
 - [7] The MPEG 4 Technology Resource
 MPEG 4 Overview
http://www.chiariglione.org/mpeg/standards/mpeg_4/mpeg_4.Htm

- [8] 寺田 努 and 塚本昌彦 and 西尾 章治郎
移動体計算環境におけるアクティブデータベースの動的トリガグラフ構築機構の設計と実装
情報処理学会論文誌：データベース，Vol. 43, No. SIG 12 (TOD 16)，pp. 52 63, 2002
- [9] Peter M. Chen and Wee Teck Ng and Subhachandra Chandra and Christopher Aycock and Gurushankar Rajamani and David Lowell
The Rio File Cache : Surviving Operating System Crashes
7 th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 74 83, 1996
- [10] C. Bobineau and L. Bouganim and P. Pucheral and P. Valduriez
PicoDBMS : Scaling down Database Techniques for the Smartcard
International Conference on Very Large Databases, No. 27, pp. 11 20, 2000
- [11] 倉光君朗 and 坂村 健
非接触型スマートカード上のコピキタスデータベース
情報処理学会論文誌：データベース，Vol. 43, No. SIG 5 (TOD 14)， pp. 110 117, 2002
- [12] H. Yokota
Autonomous Disks for Advanced Database Applications
International Symposium on Database Applications in Non Traditional Environments, pp. 441 448, 1999 November
- [13] M. Rosenblum
The Design and Implementation of a Log Structured File System

執筆者紹介 小野田 英 樹 (Hideki Onoda)
1998 年日本ユニシス(株)入社 . 都銀システムの開発業務に従事 . 2002 年 4 月より 2004 年 3 月まで奈良先端科学技術大学院大学に留学 . 現在 , 日本ユニシス・ソリューション(株)ADCoE SIS プロジェクトに所属 .