クエリプラン利用先読み技術における多重処理実行時の性能モデル検討

出射 英臣　　茂木 和彦　　西川 記史

（株）日立製作所 システム開発研究所
〒215-0013　神奈川県川崎市麻生区王禅寺 1099
E-mail: hideomi.idei.ub@hitachi.com, kazuhiko.mogi.uv@hitachi.com, norifumi.nishikawa.mn@hitachi.com

あらまし
近年、急速に普及したデータウェアハウス等の IT システムにおける各種アプリケーションの基盤として、DBMS(RDBMS)の重要性は益々高まっている。DBMS が管理するデータは年々爆発的に増加しており、大規模データベースにおける検索性能の向上は更に増して重要となっている。以上の背景から、RDBMS がクエリ実行時に作成するクエリプランを基に先読みを実施するクエリプラン利用先読み技術を研究している。本技術では、B-Tree 索引を利用した検索処理において、RDBMS がアクセスするデータを特定し、予めストレージキャッシュ上に先読みしておくことで I/O 時間の短縮を図り、DB 検索性能を向上する。今回、本技術におけるクエリ多重実行時の課題解決のため、プロトタイプにおける性能モデル構築と評価を実施し、先読み動作の最適化を実現する見通しを得た。

キーワード　ストレージ，RDBMS，クエリプラン，先読み，性能モデル

Performance Model of Prefetch Technology Using Query Plan on Multiplex Query Execution

Hideomi Idei,　Kazuhiko Mogi, and Norifumi Nishikawa

Systems Development Laboratory, Hitachi, Ltd.
1099, Ouzenji, Asao-ku, Kawasaki-shi, Kanagawa, 215-0013, Japan
E-mail: hideomi.idei.ub@hitachi.com, kazuhiko.mogi.uv@hitachi.com, norifumi.nishikawa.mn@hitachi.com

Abstract
In recent years, the importance of DBMS(RDBMS) is increasing as a base software of the various application programs in IT systems such as datawarehouses which have become into wide use. Because the amount of the data managed by the DBMS is growing explosively every year, the improvement of the performance to search the data stored in a large database is quite important. From the above background, we have studied a prefetch technique based on the query plan made by the RDBMS at the time of query execution. With this technique, the data which may be accessed by the RDBMS using B-Tree indexes are specified based on the query plan and prefetched into a storage cache in order to improve the search performance. For problem solution of prefetch technology using query plan on Multiplex Execution, We constructed a performance model and evaluated the model and got a prospect of further optimization.

Keyword　Storage, RDBMS, Query Plan, Prefetch, Performance Model

1. はじめに

近年、急速に普及したデータウェアハウス等の IT システムにおける各種アプリケーションの基盤として、DBMS（RDBMS）の重要性は益々高まっている。RDBMS が管理するデータは主に大容量のストレージに記憶されるが、そのデータ量は年々爆発的に増加しており、大量のデータの中から必要なデータを検索する処理の検索性能向上は更に増して重要となっている。

ストレージに記憶してある DB のデータにアクセスする一連の過程の中で、ボトルネックになりやすい箇所として、ストレージ内部の HDD からデータを実際に読み出す処理が挙げられる。CPU 内部の処理等が数マイクロ秒～数十マイクロ秒オーダーで完了するのに対し、HDD からのデータ読み出し処理ではシーク等の機械的な動作を伴うため、処理時間が数ミリ秒オーダーになってしまうことがその理由である。

ストレージでは、このボトルネックを解消するため、装置内に大容量のキャッシュメモリを設け、データの再利用や先読み処理によって I/O 性能を向上している。しかし、現在、この先読み処理によって得られる効果はシーケンシャルアクセスの場合に限定される。ランダムアクセスの場合、ストレージはどのデータを先読みして良いか判断する手段がなく、I/O を受け付けてからデータを読み出すことになる。

以上の背景から、当社ではストレージの高性能化を図る技術を研究している。その一技術として、RDBMS がクエリ実行時に作成するクエリプラン（どの表や索引を、どの順で、どの様にアクセスするといった実行手順に関する情報。RDBMS はこのクエリプランに沿って処理を進める。）を基に、ランダムアクセス時においても RDBMS がアクセスするデータを特定し、ストレージキャッシュ上に先読みを行うことで高性能化を図る技術（以下、クエリプラン利用先読み技術）の検討を重ね、プロトタイプの開発と評価を実施した。その結果、本技術の先読みによってクエリの実行速度を最大で 5.9 倍向上することを実証した[2]。また、現実には複数のクエリを多重実行することが多いため、クエリ多重実行に対応したプロトタイプを開発し、クエリ多重実行時においても本技術が有効であることを検証する評価を実施した[3]。

今回、クエリ多重実行時における課題解決のため、本技術のプロトタイプにおける性能モデル構築と評価を実施し、先読み動作の最適化を実現する見通しを得た。本論文では、その内容をまとめる。

RDBMS のクエリプランを利用して先読みを行う技術に関するその他の研究例として、“高機能ディスクにおけるアクセスプランを用いたプリフェッチ機構に関する評価”をあげることができる[1]。この論文では、アクセスプラン（クエリプラン）を用いたプリフェッチ（先読み）によって、どの程度の性能向上が見込まれるか調べる為にシミュレーション実験を行っている。本論文では、索引解析による先読み処理のアルゴリズムを明確にし、実機によって本機能の評価を実施している。

以下、2 章ではクエリプラン利用先読み技術の概要について、3 章では性能モデルについて、4 章では評価の結果について、5 章では本論分のまとめについて述べる。

2. クエリプラン利用先読み技術

2.1. 概要

クエリプラン先読み技術は、RDBMS がクエリ実行時に作成するクエリプランを基に索引データを取得／解析して RDBMS がアクセスするデータを特定し、それらのデータをストレージキャッシュ上に予め先読みしておくことで I/O 応答時間の短縮を図り、DB 検索性能を向上する技術である。本技術の概略を図 1 に示す。

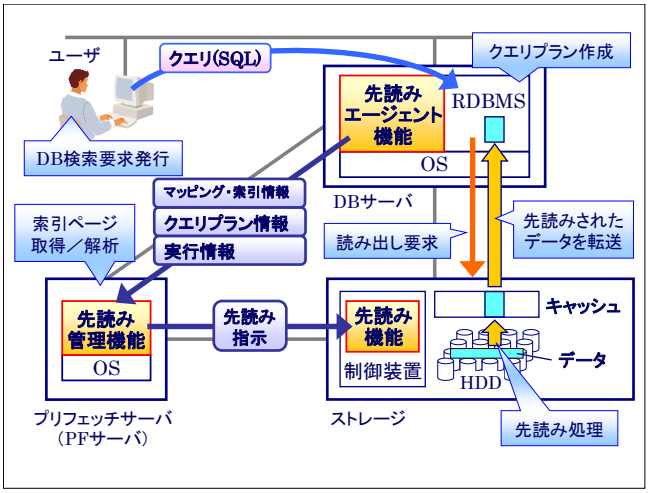

図 1　クエリプラン利用先読み技術概略

図 1 に示す様に、プロトタイプのシステムは、RDBMS が動作する DB サーバ、DB データを格納するストレージ、先読み処理の主要となるプリフェッチサーバ（以下、PF サーバ）を接続した構成である。尚、本プロトタイプで用いるストレージは、複数の HDD、大容量キャッシュメモリ、制御装置を有し、複数の HDD で LU の RAID 構成（以下、RAID グループ）が可能な高性能ストレージである。また、DB サーバ、PF サーバ、ストレージに以下 3 つの機能をそれぞれ配した機能構成となる。

(1) 先読みエージェント機能

DB サーバ上で動作する。先読み処理に必要となる情報を RDBMS や OS から取得し、PF サーバの先読み管理機能に送信する機能である。取得／送信を行う情報を表 1 に示す。

表 1　先読みエージェント機能の取得／送信情報

#	情報名	情報内容
1	マッピング情報	DB サーバにおける DB オブジェクト（表・索引）－論理ユニット（ストレージ上のデータ記憶単位）間のデータマッピングに関する情報である。先読みエージェント機能の起動時に、RDBMS や OS から取得し、先読み管理機能に送信する。
2	索引情報	索引の定義に関する情報である。先読みエージェント機能の起動時に、RDBMS から取得し、先読み管理機能に送信する。
3	クエリプラン情報	クエリプランに関する情報である。RDBMS がクエリを受け付けた際に RDBMS から取得し、先読み管理機能に送信する。
4	実行情報	RDBMS が発行した I/O 先の情報に関し、I/O 先の位置情報、及び I/O を発行したクエリを識別する情報の組みである。RDBMS がデータにアクセスする毎に RDBMS から取得し、先読み管理機能に送信する。

(2) 先読み管理機能

PF サーバ上で動作し、先読み処理を管理／実行する機能である。先読みエージェント機能から受信した情報に基づいて、索引ページを取得／解析し、RDBMS がアクセスするデータを特定した後、それらのデータの先読み指示（本プロトタイプでは SCSI プリフェッチコマンドを使用）をストレージ装置の先読み機能に送信する。

(3) 先読み機能

ストレージ装置上で動作する。先読み管理機能から受信した先読み指示に従い、対象のデータを HDD からキャッシュ上に読み出す機能である。

2.2. 先読み方式

2.2.1. 先読みの概略

クエリプラン利用先読み技術は、一連の先読みの処理過程において、先読みの対象となる複数のデータをストレージのキャッシュ上に読み出す。このデータの読み出しはストレージ内部で HDD 毎に並列に行われ、RDBMS からの次リード I/O までに対象となるデータをストレージキャッシュ上に読み出しておくことになる。その結果、DB サーバからの I/O 応答時間を短縮することが可能となる。図 2 に先読みの有無による HDD の稼動イメージを示す。

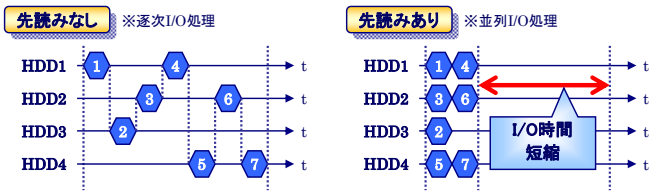

図 2　先読みの有無による HDD の稼動イメージ

2.2.2. 先読み処理の概略

PF サーバ上の先読み管理機能は、先読みエージェント機能から送信された情報に基づいて、先読みを行う処理（先読み処理）を実行する。先読み処理では、検索キー値が未知の場合の索引検索において、以下の処理を実施する。

1) DBMS がアクセスした索引リーフページを取得する。
2) 取得したリーフページを解析し、検索キー値毎にグループを作成する。
3) DBMS が表データページにアクセスした際、上記グループのいずれかに属するか判定し、属するグループがあった場合は、同グループの表データページの先読みを実施する（一次先読み）。

また、上記索引検索の後に検索キー値が既知の索引検索が続く場合、上記一次先読みの表データページを取得し、それに続く次の索引、及び表データページの先読みを実施する（二次先読み）。

性能評価で用いるベンチマーク TPC-H[4]の Query8 を例にした場合、PART 表を検索した後、その結果と Lineitem 索引によって Lineitem 表とネストループ結合する部分が一次先読みの対象範囲となり、更にその結果と Orders 索引によって Orders 表とネストループ結合する部分が二次先読みの対象範囲となる（図 3）。尚、その先は、ハッシュ結合であるため先読みの対象にはしない。

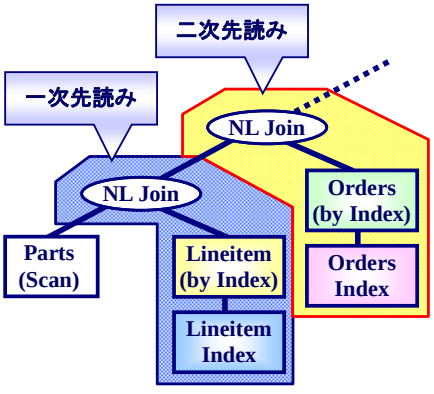

図 3　TPC-H Q8 における先読み対象部分

図 4 は、TPC-H Q8 における先読み関連ページの依存関係と先読み対象ページを示している。ある検索キー値の Lineitem 索引リーフページに対応する Lineitem 表データページが一次先読みとして実施され、更に Orders 索引ページ、及び Orders 表データページが二次先読みとして実施される。尚、先読みされるデータ量は一次先読みにおける索引のファンアウト(「索引における 1 つの検索キー値に対応する行数」と定義。Lineitem 索引の場合、平均 30)によって決まる。また、Orders 索引ページの先読みにおいて、中間ページはバッファヒットを想定して先読みを行わず、索引リーフページのみ先読みを行う。

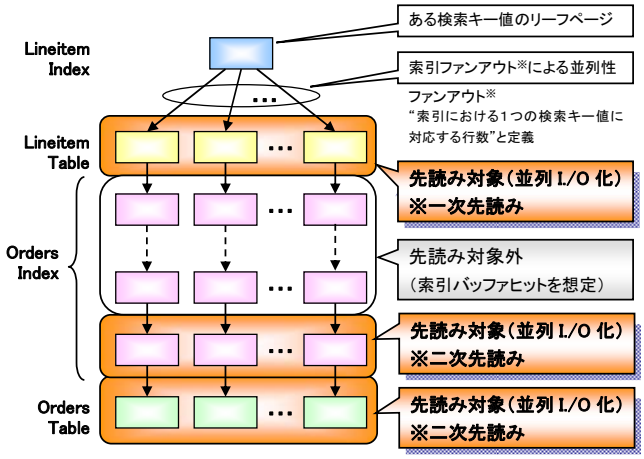

図 4　TPC-H Q8 における先読み関連ページの依存関係と先読み対象ページ

2.3. 多重処理実行時の課題

現プロトタイプにおいて、複数のクエリが並列に（多重）実行された場合、PF サーバ側では各々のクエリに対応した先読み処理を並列に実行する（図 5）。

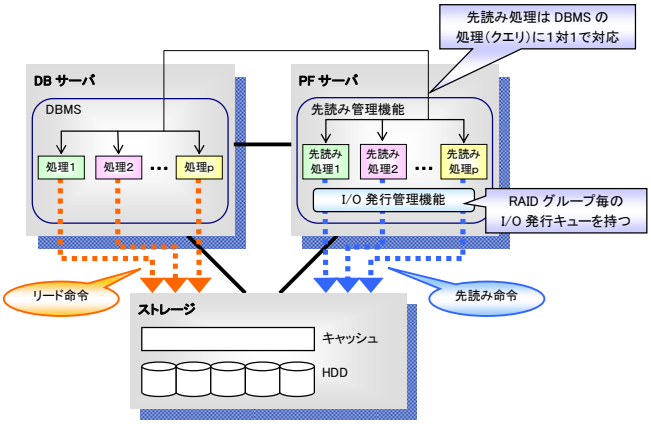

図 5　クエリ多重実行における先読み処理

これまでの評価により、クエリ多重度の増加に伴い、DB サーバからの I/O だけでなく PF サーバからの先読み I/O も増加し、先読みを実施しない場合よりも早くストレージリソースが飽和、先読み効果がその制約を受けるといった課題が発見された。

一方、現プロトタイプではクエリ多重度やストレージの負荷状況を考慮しないで先読み処理を実行し、次々と先読み I/O を発行する。そこで、クエリ高多重実行時におけるストレージの性能ボトルネックを改善する方式を検討するため、まず性能モデルを構築して評価を実施し、現プロトタイプにおける処理時間決定要因、及び処理最適化のポイントを把握する。

3. 性能モデル

3.1. 性能モデルにおける性能算出方法の概略

3.1.1. モデル化のポイント

先読み実行時における先読み周期内の処理の遷移のイメージ図を図 6 に示す。尚、先読み周期とは、一次先読みの起点となる索引リーフページの 1 ページ分のデータから先読み可能な一次先読み、及び二次先読み（二次先読みがある場合）が完了するまでの周期を言う。性能モデルを構築にするあたり、モデル化のポイントを以下に示す。

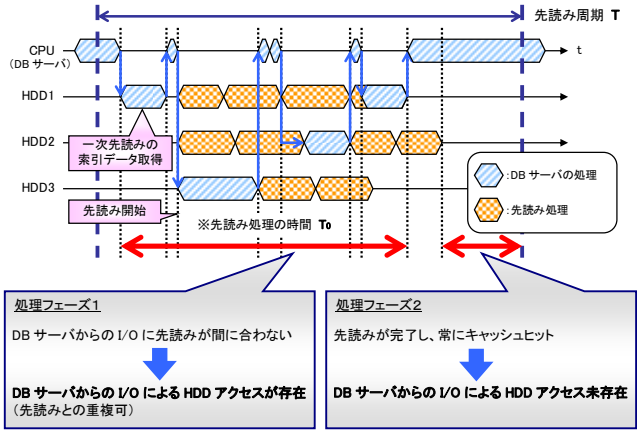

図 6　先読み実行時における処理遷移イメージ

- 性能モデルにおける処理フェーズは、先読み実施中のため DB サーバからの I/O による HDD アクセスが存在する処理フェーズ 1 と、先読みが完了し DB サーバからの I/O による HDD アクセスが未存在の処理フェーズ 2 に分類される。尚、その他の処理はこれらに比べ処理時間が十分に短いため無視できるものとする。
- 処理フェーズ 1 において、先読みが完了するまでは DB サーバからの I/O 要求がそのまま HDD アクセスとなる（キャッシュヒットしても次の I/O 要求が即時に来るため、常時 HDD にアクセス中と近似可能）。
- 処理フェーズ 2 において、先読み完了後は次の先読み周期に入るまで HDD へのアクセスは実施されない。

・ アクセス先はランダムとみなし、HDD アクセスの平均所要時間が常に一定とする。
・ 先読み管理機能における I/O は RAID グループ毎に制御される。

3.1.2. 性能算出方法の概略

性能算出方法の概略を以下に示す。

・ クリティカル HDD（確率的に最もビジーな状態にあると想定する RAID グループの HDD。但し、実際に最もビジーであるとは限らない。）において、処理可能な I/O 要求が存在しない確率（HDD アイドル率）を性能モデルから確率的に近似計算する。
・ アクセス先はランダムとみなすため、HDD アクセスの平均所要時間、HDD アイドル率からクリティカル HDD のスループットを求め、処理時間と他 HDD の稼動状態を算出する。

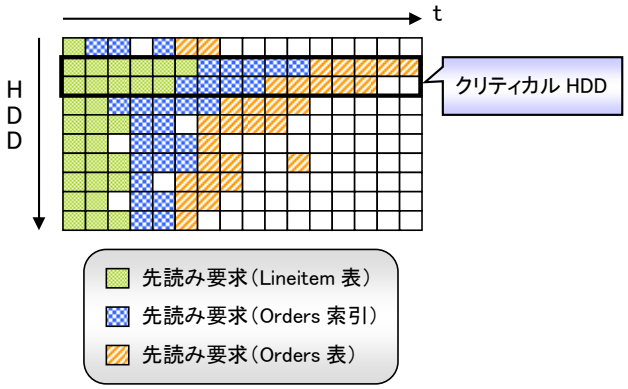

図 7　クリティカル HDD のイメージ図

3.2. 性能算出方法

3.2.1. 先読み要求が存在する確率の算出方法

常時先読み要求が存在する場合において、HDD に先読み要求が存在する確率を計算する。

(1) 1 つの処理に対して先読み処理を実施する場合

先読み完了時間は最も先読み要求が多い HDD に依存するため、先読み対象の HDD に先読み要求が存在する確率は、次の計算式により近似計算する。

先読み対象の*HDD*に先読み要求が存在する確率＝
全*HDD*の平均先読み数／*HDD*における先読み数最大値の平均値

HDD における先読み数最大値の平均値は、総先読み要求数を HDD に分散化するモンテカルロシミュレーションにより算出する。尚、クエリのアクセス特性により、HDD における先読み数最大値の平均値は変化する。

(2) 複数の処理に対して先読み処理を実施する場合

上記計算式において、対象処理の全先読み要求を考慮した場合の対象 HDD の平均先読み数・HDD における先読み数の最大値の平均値を利用する。この場合、上記先読み要求パターンの処理毎に無限繰り返しの重ね合わせに近似可能であるため、同時実行の先読み数を基にモンテカルロシミュレーションによって算出する。

3.2.2. クリティカル HDD のアイドル率算出方法

(1) 常時先読み要求が存在する条件化でのクリティカル HDD のアイドル率算出方法

p 個の処理が先読み実施中である時のクリティカル HDD アイドル率 P_{idle} は、次の計算式で求まる。

$$P_{idle}(p)=\sum_{i=0}^{p}(i\text{個の処理が先読み処理 実施中})\times(i\text{個の処理が先読み実施 中だが}HDD\text{はアイドルの確率})$$

ここで、先読み周期時間を T、先読み周期における先読み処理中の時間（図 6 で示した処理フェーズ 1 に相当）を T_0 とした場合、i 個の処理が先読み実施中の確率は、次の計算式で求まる。

$$i\text{個の処理が先読み実施中の確率}={}_pC_i\bullet\left(1-T_0/T\right)^i\bullet\left(T_0/T\right)^{p-i}\quad\text{【ランダム近似】}$$

また、先読み処理が一次先読みの索引データ取得とデータ先読み実施中に分離されるため、P_{idle} は、次の計算式に近似可能となる。

$$P_{idle}(p)=\left(T_0/T\right)^p+\left[\sum_{i=0}^{p}{}_pC_i\bullet\left(1-T_0/T\right)^i\bullet\left(T_0/T\right)^{p-i}\right.$$
$$\bullet(1-i\text{個の処理による}DB\text{サーバからの}I/O\text{要求存在確率})$$
$$\left.\bullet\sum_{j=0}^{i}\left((j\text{個の処理がデータ先読み実施中の確率})\right.\right.$$
$$\left.\left.\bullet(1-j\text{個の先読み処理による}I/O\text{要求存在確率})\right)\right]$$

(2) クリティカル HDD のアイドル率算出に必要となる確率の計算方法

i 個の処理が先読み処理実施中において、DB サーバからの I/O 要求の存在確率は、次の計算式で求まる。尚、本計算式における 1 つの処理が対象 HDD をアクセスする確率は、アクセスが完全なランダムの場合、一般的な確率で計算可能であり、アクセスに偏りがある場合はモニタ等で採取したデータを基に計算可能である。今回は、両者で計算した結果、どちらも大差がなかったため、確率で計算した値を使用する。

*i*個の処理による*DB*サーバからの*I/O*要求の存在確率＝

$1-(1-1つの処理が対象HDDをアクセスする確率)^{i}$

i 個の処理がデータ先読み中の確率は、次の計算式に近似可能である。

*i*個の処理のうち*j*個の処理がデータ先読 み中の確率＝

$${}_{i}C_{j} \bullet \left(\left(T-T_0-T_2\right)/\left(T-T_0\right)\right)^{j} \bullet \left(T_2/\left(T-T_0\right)\right)^{i-j} \quad 【ランダム近似】$$

上記計算式では一次先読み索引データ取得時間を T_2 とし、その値は、一次先読み索引データ取得時の対象 HDD の先読み要求存在確率を考慮した次の計算式から近似計算する（尚、処理多重度を p、1HDD あたりのアクセス時間平均値を T_{HDD} とする）。

$$T_2=(1+2\alpha)\bullet T_{HDD} \quad , \quad \alpha=1-\left(1-\left(T_0+T_2\right)/T\right)^{p-1}$$

j 個の先読み処理による I/O 要求が存在する確率は、3.2.1 章(1)で述べた先読み要求が存在する確率（モンテカルロシミュレーションによる計算値）を利用する。

3.3. 処理時間の算出方法

先読み対象の HDD が性能ボトルネックとなっている（CPU 待ち時間は無視できる）ため、クリティカル HDD の処理スループットから処理時間の算出が可能となる。クリティカル HDD のスループットは、I/O 平均処理数と HDD ビジー率(=1-P_{idle} (p))から算出する。処理多重度を p、1HDD あたりのアクセス時間平均値を T_{HDD} とした場合、処理時間は、次の計算式で求まる。

処理時間＝

$$\frac{p\times(1処理あたりのクリティカルHDDで処理するI/O数の平均値)\times T_{HDD}}{1-P_{idle}(p)}$$

尚、HDD ビジー率、後述の評価環境で測定した処理時間の実測値を基に T_{HDD} を計算した場合の誤差は最大で 3%となった（但し、1 多重時を除く。1 多重の場合、最大 11.8%の誤差があり）。従って、予測精度を判断する際、HDD ビジー率の誤差をもって性能モデルの正しさを判断する。

4. 評価

4.1. 評価環境

処理モデル検証のため、実測値との比較を実施する。実測値の測定に用いたプロトタイプのシステム構成は、DB サーバー-PF サーバ間を 100Mbps Ethernet と FC(Fibre Channel)で接続し、DB サーバーストレージ間、及び PF サーバーストレージ間を FC で接続したシステム構成である。また、I/O トレース採取のため、PF サーバ、DB サーバとストレージ間に FC もモニタを接続する。プロトタイプのシステム構成を図 8、構成機種を表 2 に示す。

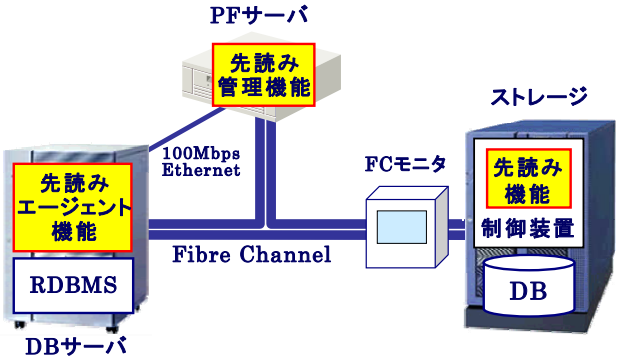

図 8　プロトタイプのシステム構成

表 2　プロトタイプのシステム構成機種

【DB サーバ】	
プロセッサ	Pentium3™ 1.13GHz
OS	Linux™ (Kernel 2.4.24)
RDBMS	HiRDB™ Version7 ※先読みエージェント I/F 拡張版
【PF サーバ】	
プロセッサ	Xeon™ 3.06GHz
OS	Linux™ (Kernel 2.4.24)
【ストレージ装置】	
機種	RAID ディスク装置 ※SCSI プリフェッチコマンド対応
キャッシュ容量	2GByte
LU 構成	RAID5(4D＋1P)×5 グループ （HDD 台数 25 台）

4.1.1. DB 環境

DB として、TPC-H (Scale Factor=3、データ量（表データのみ）約 3GByte、索引ページサイズ 4KByte、表データページサイズ 16KByte)の環境を使用し、データの配置として、次の 2 パターンを用意した。

(1) データ配置 A

同一の先読み周期内において、ある Lineitem 表のデータに対応して引き続きアクセスされる Orders 索引、Orders 表のデータのうち 92%が元の Lineitem 表データと同一の RAID グループに存在するデータ配置。

(2) データ配置 B

同一の先読み周期内において、ある Lineitem 表のデータに対応して引き続きアクセスされる Orders 索引、Orders 表の各データが 92%以上の確率で異なる RAID グループに存在するデータ配置。

4.2. 性能モデルの検証

性能モデルの検証において、前述のプロトタイプで採取した I/O トレースの解析結果を基に算出した値をモデルパラメータとして、性能モデル値を算出する。用いたモデルパラメータは「T_0=54.2[ms]、T_{HDD}=5.19[ms]、HDD 台数=25（5RAID グループ）、先読み並列度=30、Orders 索引データへのアクセスは 73%の割合で中間ノードデータを取得し、索引データの 46%、表データの 7%が先読みなしの場合でキャッシュヒットする」である。

4.2.1. HDD ビジー率平均値の比較

HDD ビジー率平均値の性能モデル値と実測値の比較結果を図 9 に示す。図 9 のグラフにおいて、横軸は処理（クエリ）多重度、縦軸は HDD ビジー率である。

性能モデル値と実測値の比較の結果、HDD ビジー率平均値において、両者の間には若干の誤差が存在する。性能モデルに用いたクリティカル HDD は、確率的に最もビジーな状態にある RAID グループの HDD であるが、ストレージ内の HDD のモニタ情報を調査した結果、各々の HDD のビジー率にばらつきが生じていた。そこで、HDD ビジー率の最大値の平均について次章で比較した。

また、処理時間の性能モデル値と実測値の比較結果を図 10 に示す。図 10 のグラフにおいて、横軸は処理多重度、縦軸が処理時間[sec]である。16 多重までは性能モデル値と実測値はほぼ同一であるが、それ以上は性能モデル値の方が若干良い値となっている。

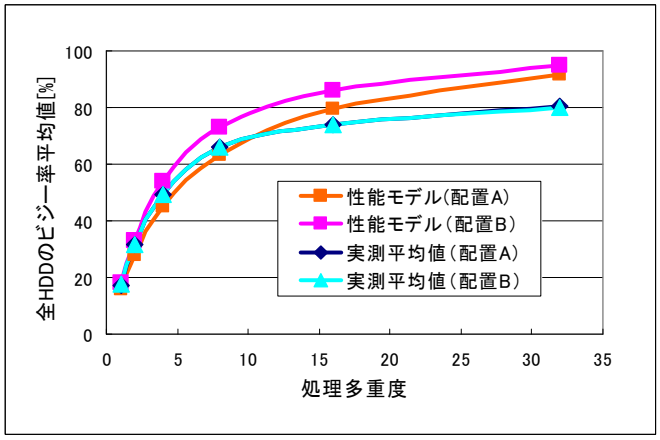

図 9　HDD ビジー率平均値の比較

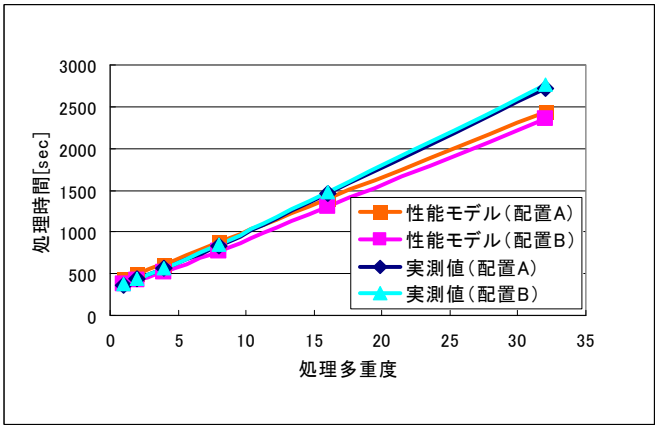

図 10　処理時間の比較

4.2.2. HDD ビジー率最大値の平均の比較

HDD ビジー率最大値の平均の性能モデル値と実測値の比較結果を図 11 に示す。図 11 のグラフにおいて、横軸は処理多重度、縦軸は HDD ビジー率である。

性能モデル値と実測値の比較の結果、16 多重までは性能モデル（データ配置 B）で算出したモデル値と実測値がほぼ一致している。16 多重以上に差が表れる原因については次章で評価する。

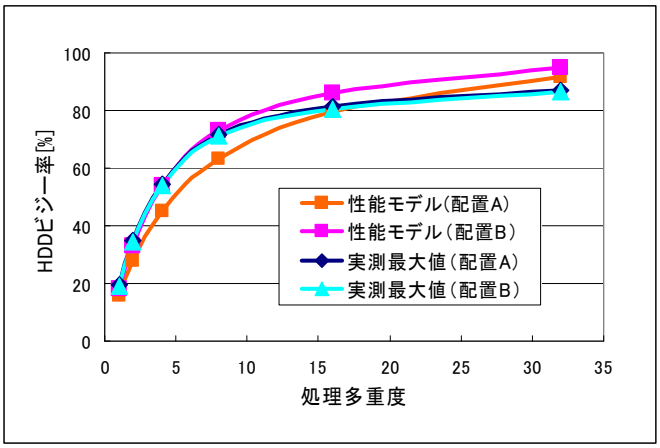

図 11　HDD ビジー率最大値の平均の比較

4.2.3. 先読み I/O 発行制御の有無による HDD ビジー率の比較

PF サーバ上の I/O 発行管理機能は、図 5 で示した様に RAID グループ毎の I/O 発行キューを持ち、閾値（現在 10）として設定された値以上の I/O を発行しないように制御をかけている。そこで、先読み I/O 発行制御によって先読み要求の存在確率が 1-(1-1/25)^10≒0.34 低下すると仮定してデータ配置 B における性能モデル値を再算出し、実測値と比較した。その結果を図 12 に示す。図 12 のグラフにおいて、横軸は処理多重度、縦軸は HDD ビジー率である。また、PF サーバ内の I/O

発行待ちキュー長の平均値を図 13 に示す。図 13 のグラフにおいて、横軸は処理多重度、縦軸は I/O 発行待ちキュー長平均値である。

性能モデル値と実測値の比較の結果、32 多重では若干性能モデル値の方が実測値より悪い結果であるもののほぼ一致した結果と言える。32 多重の場合には、図 13 のグラフにあるように、PF サーバ内の I/O 発行待ちキュー長が大幅に上昇し、これが先読み I/O 発行制御に影響を与え、上記の差に現れたと考えられる。但し、性能への影響は少なく誤差の範囲と言える。

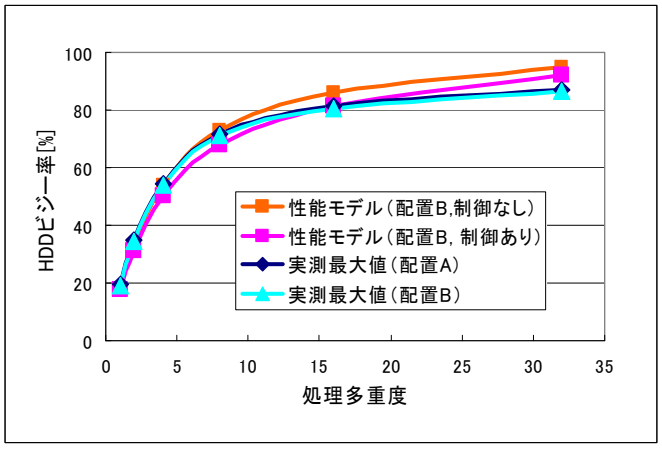

図 12　先読み I/O 発行制御の有無による HDD ビジー率最大値の平均の比較

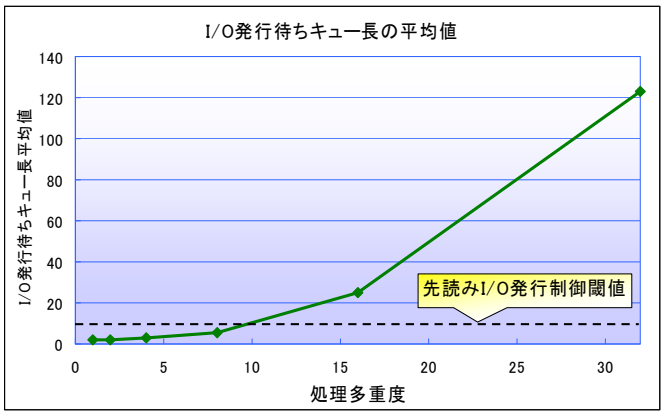

図 13　PF サーバの I/O 発行待ちキュー長の平均値

4.3. 先読み動作の最適化方式案

クエリの高多重実行時（ストレージ過負荷時）における性能改善方式として、先読みの有無による HDD ビジー率の差を基に先読み I/O 発行可能数を計算し（先読みなしの場合の HDD ビジー率はランダム I/O の存在確率として算出）、PF サーバ側で先読み I/O を制御する方式が考えられる。これにより、ストレージ負荷を適切な範囲に収めることが可能になると考えられる。

5. まとめ

現在、RDBMS がクエリ時に作成するクエリプランを基に先読みを実施するクエリプラン利用先読み技術を研究している。本技術のプロトタイプにおいて、複数のクエリが並列に（多重）実行された場合、DB サーバからの I/O だけでなく PF サーバからの先読み I/O も増加し、先読みを実施しない場合よりも早くストレージリソースが飽和して先読み効果がその制約を受けるといった課題がある。一方、現プロトタイプではクエリ多重度やストレージの負荷状況を考慮しないで先読み処理を実行し、次々と先読み I/O を発行している。そこで、上記課題解決のため、プロトタイプの性能モデル構築と評価を実施し、先読み動作の最適化を実現する見通しを得た。

謝辞

本研究に関して御指導いただきました東京大学の喜連川教授に深く感謝致します。

本論文で記された技術には、文部科学省が実施するリーディングプロジェクト「e-Society 基盤ソフトウェアの総合開発」のストレージ・データベース融合技術（東大，日立）で技術開発された成果が反映されています。

参考文献

[1] 向井景洋，根本利弘，喜連川優，“高機能ディスクにおけるアクセスプランを用いたプリフェッチ機構に関する評価”，DEWS00 講演論文集 3B-3

[2] 出射英臣，茂木和彦，西川記史，大枝高，“クエリプランを利用した先読み技術の開発と初期評価”，DEWS2005 講演論文集 5B-01

[3] 出射英臣，茂木和彦，西川記史，大枝高，“クエリプランを利用した先読み技術のクエリ多重実行時における性能の評価”，情報処理学会 第 68 回全国大会 5D-5

[4] TPC BENCHMARK™ H Standard Specification Revision 1.3.0 仕様書　http://www.tpc.org/tpch