

データに基づく品質管理を実現するためのデータベース物理設計の評価

西川 記史[†] 藤原 真二^{††} 早水 悠登^{†††} 合田 和生^{†††}

[†] 株式会社 日立製作所 〒185-8601 東京都国分寺市東恋ヶ窪一丁目 280 番地

^{††} 株式会社 日立製作所 〒224-0817 神奈川県横浜市戸塚区吉田町 292 番地

^{†††} 東京大学 生産技術研究所 〒153-8505 東京都目黒区駒場 4-6-1

E-mail: [†]{norifumi.nishikawa,mn,shinji.fujiwara.yc}@hitachi.com, ^{††}{haya,kgoda}@tkl.iis.u-tokyo.ac.jp

あらまし 製品に対する安全意識の高まりから、製造業を中心にデータに基づく品質管理が重要となっている。データに基づく品質管理では、集計処理に加え、製造工程に合わせてデータをトレースする処理が行われるが、分析用データベースの物理設計の主流となっているカラムストアとレンジ索引の組合せではデータのトレース処理に十分対応することができない。本論文では、製造業データベースのベンチマークを用い、複数の物理設計のパートナーについて、データの集計およびトレース処理の性能とデータベースの容量を評価した。評価の結果、集計処理ではレンジ索引が、データのトレース処理では B-tree 索引とレンジ索引の組合せが有効であること、カラムストアが性能向上と容量削減に有効であることを確認した。これにより、データに基づく品質管理を実現できる見込みを得た。

キーワード データ構造・索引、ストレージ管理、スマートファクトリー、IoT、トレーサビリティ、品質管理

1 はじめに

近年、製品に対する安全意識の高まりから、様々な分野でデータに基づく品質管理が重要となっており、流通・小売、物流、製造業など多くの分野でデータに基づく品質管理が行われている [1] [2] [3]。データに基づく品質管理では、製造データを履歴として蓄積しそれを用いて様々な分析処理を行う。製造データは、4M モデル [4] に代表されるグラフ構造のデータを含む。また、分析処理は集計処理に加え、グラフ構造データを辿る処理（トレース処理）を含む。

データベースの物理設計はデータベースの性能に多大な影響を与えることが知られている。従来、分析処理に用いられるデータベースの物理設計として、カラムストアとレンジ索引が広く用いられている [5] [6]。これらの物理設計は大規模なデータの集計などには適しているものの、少数のデータの選択を再帰的行うデータのトレース処理には不向きであると考えられる。しかし、前述の製造データのトレースのようなグラフ構造のデータを効率よく探索するための物理設計については十分な検討がなされていない。

これまで、著者らはデータベースの物理設計がグラフ構造データのトレース処理にどのような影響を与えるかについて初期評価を行っており [7]、トレース処理においてはカラムストアと B-tree 索引の組合せがデータベースの容量削減と性能向上に有効であることを示してきた。

本論文では、上記の研究を一步進め、データベースの物理設計の様々な組み合わせについて、著者らが開発した製造業データベースのワークロードを模したベンチマークである 4mbench ベンチマーク [8] を用いてどのような物理設計が集計処理やトレース処理を含む製造業データベースのワークロードに適しているかを評価する。評価に用いた物理設計の組合せは、テーブ

ルデータの格納方法としてローストア及びカラムストア、索引として B+Tree 索引 [9]、レンジ索引 [10]、及び索引無しである。評価の結果、集計処理ではレンジ索引が、データのトレース処理では B-tree 索引とレンジ索引の組合せが有効であること、カラムストアが性能向上と容量削減に有効であることを確認した。この結果、適切な物理設計を選択することにより、データに基づく品質管理を実現できる見込みを得た。

以下、2 章ではデータに基づく品質管理について、3 章では我々が開発している 4mbench ベンチマークの概要について述べる。4 章では 4mbench を用いたデータベースの物理設計の評価、5 章では関連研究について述べ、最後に 6 章でまとめる。

2 データに基づく品質管理

製造業の製造現場で取得されるデータを工場データと呼ぶが、工場データのうち設備により取得されるデータは設備データと製造データに大別できる [11]。設備データとは設備の温度や振動など一定間隔で取得される時系列データであり、製造データとは製造条件や使用した部品ロットなど製造時に取得されるデータである。

データに基づく品質管理は、前述の設備データと製造データを用いた製品の品質管理である。これらのデータを蓄積し追跡可能な状態にすることで、製品品質に問題があった場合に製品から部品や材料ロットを特定するトレースバックや生産設備・機器にエラーが発生した際に影響範囲を特定するトレースフォワードなどを迅速に行うことが可能となる [12]。

3 製造業データベースベンチマーク 4mbench

3.1 4M モデルによる製造工程のモデル化

4M モデルは、主に製造業の製造工程を表すために用いられる

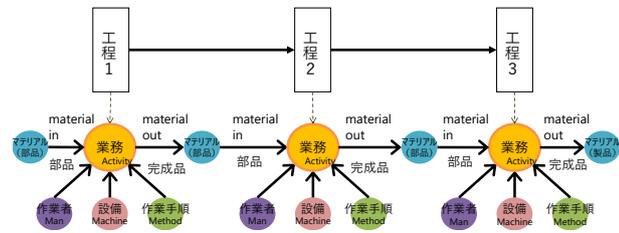


図 1 4M を用いた製造工程のモデル化

データモデルであり、製造工程を huMan, Machine, Material, 及び Method という 4つの観点でモデル化している。製造工程には業務が対応し、業務は 1つ以上のマテリアル(部品または原材料)を入力し、加工を行ったのち 1つ以上の Material(製品)を出力する。また、業務には製造を行った作業員(huMan)、製造を行った設備(Machine)、作業手順(Method)が対応付く。そして、マテリアルを介して各業務を関連付ける。図 1 に、4M モデルに基づく製造工程のモデル化を図示する。

3.2 4mbench の概要

4mbench ベンチマーク [8] は、製造業で用いられるワークロードを模したベンチマークである。4mbench で定義されているクエリは、時間当たりの製品の生産数や設備稼働率、歩留まり算出など統計的な処理を行う統計的クエリが 3 個、出荷判定や機器不良時の影響・原因調査などの工程データをトレースするトレースクエリが 3 個の計 6 個である。

データベースとしてリレーショナルデータベースを想定しており、前述の 4M の構成要素をスキーマとして保持している。具体的には、製造を行った人、製造を行った機器、製造方法を格納する 3 種類のディメンジョンテーブルと、利用された部品または原材料と生成された製品(マテリアル)の履歴、それらを生成した業務の履歴、および機器の稼働履歴を格納する 3 種類のファクトテーブルを含む。そして、マテリアルが自身を生成した業務、利用した業務を介してグラフ構造として接続されている。4mbench ベンチマークの実装 [14] で入手可能である。

3.3 4mbench クエリ

本節では、4mbench で実装されている統計的クエリとトレースクエリの典型的な例について説明する。

3.3.1 統計的クエリ

統計的クエリの一例は生産量の分析であり、ある中間品および最終製品が一定時間内に生成された個数を求めるものである。これは、対象とする期間に生成されたマテリアルをその種類ごとに集計し個数を求めるものであり、従来の分析クエリと同等のクエリである。

3.3.2 トレースクエリ

トレースクエリの一例は製品出荷時の品質テストクエリである。このクエリは完成品について、まずその完成品とそれらを構成する全ての部品(マテリアル)について、製品または部品を生成した業務が、部品の出力先の業務と等しい部品を探索する。次に探索した部品を入力とする業務(製品または探索した部品から生成されている部品を生成した業務)、その業務に用

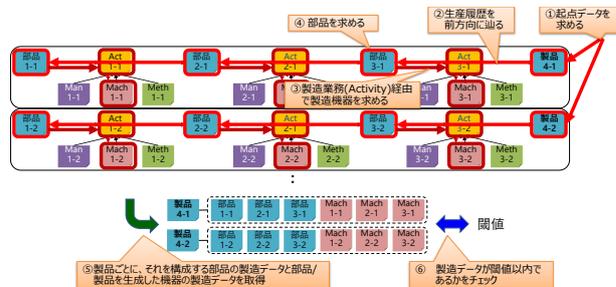


図 2 品質テストクエリの動作

いた機器の情報を取得する。そして完成品を構成する全ての部品が適切な条件で製造されているか、製造された中間品の製造値が適切であるかをチェックする。この例を図 2 に示す。

このような種類の分析クエリは TPC-H [13] のような分析処理のベンチマークには含まれておらず、4mbench を特徴付けるものとなっている。4mbench におけるトレースクエリは、SQL の再帰処理を用いて実現されている。

4 4mbench を用いたデータベースの物理設計の評価

本章では、4mbench を用いて、製造業向けデータベースの物理設計による性能と容量の差異を比較した結果を示す。我々は、商用の RDBMS である Hitachi Advanced Data Binder(HADB) 上で 4mbench ベンチマークを稼働させ、クエリ性能とデータベース容量を評価した。

4.1 評価条件

4.1.1 機器構成

評価に用いたサーバは HA8000V/DL380 Gen 10 であり 2 つの Intel Xeon Gold 6240R (24 コア) CPU, 768GB メモリを搭載している。OS は Red Hat Enterprise Linux 8.3 である。ストレージは Hitachi Virtual Storage Platform E790 であり、32Gb ファイバチャネル 2 本でサーバと接続されている。

ストレージは 7 台の 2TB の NVMe SSD を用いた RAID-5 RAID グループが 6 台あり、それぞれから 2TB の LU を 4 つ、合計 24 個の LU を切り出してサーバに提供している。サーバでは Linux LVM を用いてこれら 24 個の LU を有するボリュームグループを構築し、そこから 8 個の LV を切り出して HADB に割り当てている。HADB では LV を 4 つずつ束ねて 2 つの DB エリアを作成し、それぞれ表用と索引用の DB エリアとして割り当てた。またサーバには 5.8TB の NVMe-SSD が 8 台搭載されており、これらを LVM で束ねたのち 10TB の LV を切り出して HADB の作業表領域に割り当てた。

4.1.2 データとクエリ

4mbench にはスケールファクタとして工場のラインの数とデータを蓄積する日数の 2 つのスケールファクタがある。我々は工場のライン数 30, 日数 120 日のデータを用いた。データ容量は非圧縮の CSV 形式で約 1,086GB である。

クエリは 3.2 節で述べた 6 種類のクエリを用いた。クエリには 4mQ.1 から 4mQ.6 の番号が付与されている。このうち

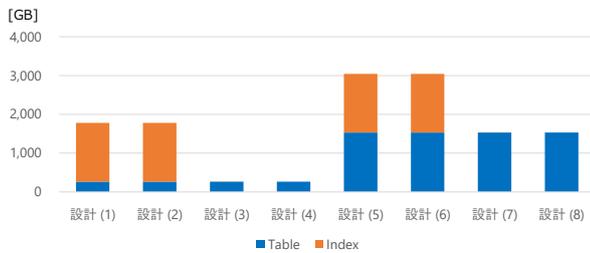


図3 データベース容量の比較

4mQ.1, 2, 6 が集計処理を行うクエリ, 4mQ.3, 4, 5 が製造データのトレースを行う再帰クエリである。

4.1.3 データベースの物理設計

データベースの物理設計としては、表データの格納形式としてカラムストアとロースタ形式を、索引として B-tree 索引、レンジ索引の組合せ (索引なし含む) を評価した。具体的には (1) カラムストア・B-tree 索引・レンジ索引, (2) カラムストア・B-tree 索引, (3) カラムストア・レンジ索引, (4) カラムストア・索引なし, (5) ロースタ・B-tree 索引・レンジ索引, (6) ロースタ・B-tree 索引, (7) ロースタ・レンジ索引, (8) ロースタ・索引なし, の 8 パターンを評価した。

4.2 評価結果

4.2.1 データベース容量の評価

我々は前述の 8 種類の物理設計について、まずデータベースの容量を比較した。その結果を図 3 に示す。図から分かるように、カラムストアの表容量はロースタの約 1/5 であることが分かる。これはカラムストアではデータが圧縮されて格納されるためである。索引容量は、B-tree 索引の容量が大きいが分かる。レンジ索引の容量は非常に小さい。

4.2.2 クエリ性能の評価

次に、クエリ性能の評価を行った。この際、48 スレッドによるアウトオブオーダー実行 [15], [16] を有効とした。この結果を図 4 に示す。図 4 は、横軸がクエリ番号、縦軸がログスケールで示された応答時間である。

まず集計クエリ (4mQ.1, 2, 6) について見てみる。これらのクエリでは、カラムストア、ロースタ共に、レンジ索引を定義した設計 (設計 (1), (3), (5), (7)) の応答時間が短いことが分かる。これはレンジ索引によりスキャンを行う範囲が削減されているためである。また、カラムストア (設計 (1)~(4)) とロースタ (設計 (5)~(7)) を比較するとカラムストアの方が高速であることが分かる。これはカラムストアの方が必要な行のみ読み込むこと及びデータが圧縮されていることにより、データ読み込み量が小さいためである。

次に再帰クエリ (4mQ.3, 4, 5) について見てみる。再帰クエリでは B-tree 索引の有無により性能が大きく異なっており、B-tree 索引を定義した設計 (設計 (1)(2)(5)(6)) の応答時間が短いことが分かる。これは、製造データのトレースでは選択率が小さい範囲を再帰的に辿るためであり、このような処理に B-tree が適していることを示していると言える。また、設計 (1) と (2), 設計 (5) と (6) を比較するとそれぞれ設計 (1)(5) が

(2)(6) より応答時間が短いことが分かる。設計 (1)(5) と (2)(6) の差異はレンジ索引の有無 (設計 (1)(5) にはレンジ索引が定義されている) であるが、これは B-tree 索引とレンジ索引の併用がより効果的であることを示している。

4.2.3 B-tree 索引とレンジ索引の組合せの効果

B-tree 索引とレンジ索引の組合せの効果を調べるために、カラムストアと B-tree 索引及びレンジ索引を用いた設計 (1) とカラムストアと B-tree 索引を用いた設計 (2) について、クエリ 4mQ.3 の論理 I/O 数の関係を調査した。この結果を図 5 に示す。

図から分かるように、B-tree 索引とレンジ索引の両方を定義した設計は、B-tree 索引のみを定義した設計と比較し、論理 I/O 回数が約 1/4 となっていることが分かる。HADB にはチャンクと呼ばれる論理構造を用いて表及び B-tree 索引を分割して管理している。レンジ索引により表及び索引をスキャンする際にデータが含まれていないチャンクをスキップすることができ、結果として論理 I/O 数を低減できたと考えられる。

これらの結果から、集計処理ではレンジ索引が、データのトレース処理では B-tree 索引とレンジ索引の組合せが有効であること、カラムストアが性能向上と容量削減に有効であると言える。

5 関連研究

データベースの物理設計についての議論には、文献 [17] [18] がある。文献 [17] では、カラムストアの格納方法をロースタ上でエミュレートした場合、タプル再構築のコストとタプルごとのオーバーヘッドにより必ずしもよい性能をもたらさないこと、カラムストアが行の実体化の遅延と圧縮により、データウェアハウスのベンチマークでは高性能となることを示している。また、文献 [18] はカラムストア索引と B-tree 索引を用いたハイブリッド構成について議論し、選択率が小さいクエリでは B-tree 索引が、カラムストア索引は大規模なスキャンや一括更新に適していることを示している。特にプライマリ B-tree とセカンダリカラムストア索引の組合せが OLTP と分析処理の共存において高速化を達成できることを示している。しかし、これらの文献には前述の製造データのトレースのようなグラフ構造のデータを効率よく探索するための物理設計については触れられていない。

6 まとめ

本論文では、製造業データベースのワークロードを模した 4mbench ベンチマークを用いて、表データの格納方法としてカラムストアとロースタ、索引として B-tree 索引、レンジ索引の組合せについてデータベースの容量とクエリの性能の評価を行った。この結果、データベースの容量はカラムストアが容量が小さいこと、また集計処理ではレンジ索引が、データのトレース処理では B-tree 索引とレンジ索引の組合せが有効であること、カラムストアが性能向上と容量削減に有効であることを示した。これらの結果から、カラムストアと B-tree 索引お

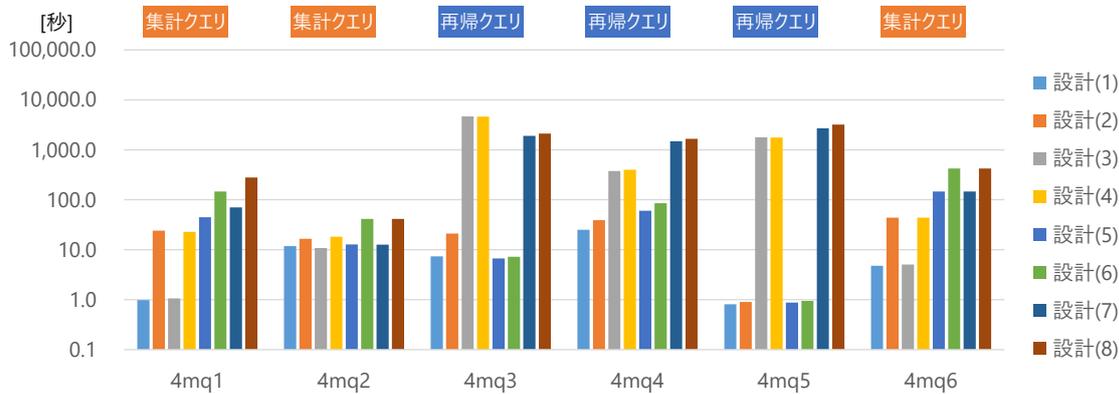


図 4 クエリ性能の比較

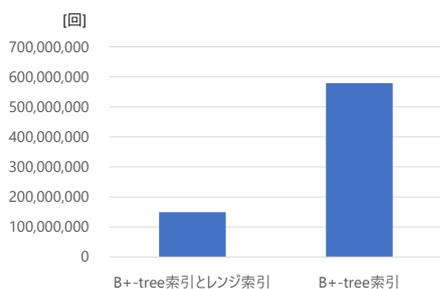


図 5 4mQ.3 の論理 I/O 数の比較

およびレンジ索引の組合せが、製造業のワークロードに適しておりデータに基づく品質管理を実現できる見込みを得た。

文 献

- [1] 宇田川 佳久, トレーサビリティの定義・効果・課題に関する考察, 情報処理学会研究報告 2009 年度 (6), 1-8, 2010.
- [2] W. Chen, Y. Liu, C. Chi, Innovative Traceability Application in Medical Devices Industry Using the Identification and Resolution System for Industrial Internet, International Conference on Measuring Technology and Mechatronics Automation (ICMTMA 2021), pp.72-76, 2021.
- [3] Q. Xiu, K. Muro, Robust Inference Traceability Technology for Product Quality Enhancement, International Conference on Industrial Engineering and Engineering Management (IEEM 2017), pp.1699-1703, 2017.
- [4] C. Favi, M. Germani, M. Marconi, A 4M Approach for a Comprehensive Analysis and Improvement of Manual Assembly Lines, Procedia Manufacturing Volume 11, pp.1510-1518, 2017.
- [5] N. Armenatzoglou, S. Basu, N. Bhanoori, M. Cai, N. Chaninani, K. Chinta, V. Govindaraju, T. J. Green, M. Gupta, S. Hillig, E. Hotinger, Y. Leshinsky, J. Liang, M. McCreedy, F. Nagel, I. Pandis, P. Parchas, R. Pathak, O. Polychroniou, F. Rahman, G. Saxena, G. Soundararajan, S. Subramanian, D. Terry, Amazon Redshift Re-invented, International Conference on Management of Data (SIGMOD 2022), pp.2205-2217, 2022.
- [6] B. Dageville, T. Cruanes, M. Zukowski, V. Antonov, A. Avanes, J. Bock, J. Claybaugh, D. Engovatov, M. Hentschel, J. Huang, A. W. Lee, A. Motivala, A. Q. Munir, S. Pelley, P. Povinec, G. Rahn, S. Trantafyllis, P. Unterbrunner, The Snowflake Elastic Data Warehouse, International Conference on Management of Data (SIGMOD 2016), pp.215-226, 2016.
- [7] 小澤麻由子, 西川記史, 渡辺聡, 坂庭秀紀, 茂木和彦, 大規模グラフ構造データの大规模トラバースに適したデータ格納方式の検討と初期評価, 第 14 回データ工学と情報マネジメントに関するフォーラム, G34-3, 2022.
- [8] K. Goda, Y. Hayamizu, N. Nishikawa, S. Fujiwara, 4mbench: Performance Benchmark of Manufacturing Business Database, TPC Technology Conference on Performance Evaluation & Benchmarking (TPCTC 2022), 2022.
- [9] R. Bayer, E. McCreight, Organization and Maintenance of Large Ordered Indices, ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control, pp.107-141, 1970.
- [10] J. Arulraj, J. Levandoski, U. F. Minhas, P. Larson, BzTree: A High-Performance Latch-free Range Index for Non-Volatile Memory, Proceedings of the VLDB Endowment, Vol. 11, No. 5, pp.553-565, 2018.
- [11] 上田宜史, 平井規郎, 生方康友, 製造データ活用による試験判定基準の最適化—量産工場の製造データ分析の事例から—, 情報処理学会デジタルプラクティス Vol.8, No.3, pp.210-217, 2017.
- [12] 株式会社日立製作所, 生産現場デジタルツイン化ソリューション「IoT コンパス」, <https://www.hitachi.co.jp/products/it/IoT/M2M/list/iotcompass/>, 2021.
- [13] Transaction Processing Performance Council: TPC-H benchmark specification, <https://www.tpc.org/tpch/>
- [14] K. Goda, Y. Hayamizu, 4mbench: a tool for performance benchmark of manufacturing business database, <https://github.com/dbc-utokyoiois/4mbench/>, 2022.
- [15] 喜連川優, 合田和生, アウトオブオーダー型データベースエンジン OoODE の構想と初期実験, 日本データベース学会論文誌, Vol.8, No.1, pp.131-136, 2009.
- [16] K. Goda, Y. Hayamizu, H. Yamada, M. Kitsuregawa, Out-of-order Execution of Database Queries, Proceedings of the VLDB Endowment, Vol. 13, No. 12, pp.3489-3501, 2020.
- [17] D. J. Abadi, S. R. Madden, N. Hachem, Column-Stores vs. Row-Stores: How Different Are They Really?, International Conference on Management of Data (SIGMOD 2008), pp.967-980, 2008.
- [18] A. Dziejczak, J. Wang, S. Das, B. Ding, V. R. Narasayya, Columnstore and B+ tree - Are Hybrid Physical Designs Important?, International Conference on Management of Data (SIGMOD 2018), pp.177-190, 2018.