

# 面向列的内存访问研究综述<sup>①</sup>



陈 瑞<sup>1,2</sup>, 夏承遗<sup>1,2</sup>

<sup>1</sup>(天津理工大学 天津市智能计算与软件新技术重点实验室, 天津 300384)

<sup>2</sup>(天津理工大学 学习型智能计算教育部工程研究中心, 天津 300384)

通讯作者: 夏承遗, E-mail: cyxia@email.tjut.edu.cn

**摘 要:** 在大规模、数据量密集的特定应用场景下, 以行存储访问数据的方式弊端日益凸显, 逐渐不能满足数据高速访问的性能需求, 数据亟需更加高效的传输和处理方式. 因此, 拓展新的内存访问方式, 并且同时兼容行、列方向的访问对提升访问效率、降低整体功耗、节省内存空间有着重要意义. 本文围绕动态随机存储和非易失性存储两个方面来详细介绍实现列方向的内存访问方式, 重点分析了存储单元的结构设计以及实现列向存储访问过程. 最后, 对内存两种不同访问方式进行了比较和总结, 并且对行列访问的内存数据库、数据挖掘、数据加密算法、实时系统的应用场景进行了展望.

**关键词:** 内存; 存储结构; 数据访问; 非易失性; 列向访问

引用格式: 陈瑞, 夏承遗. 面向列的内存访问研究综述. 计算机系统应用, 2021, 30(12): 10-17. <http://www.c-s-a.org.cn/1003-3254/8252.html>

## Survey on Column-Oriented Memory Access

CHEN Rui<sup>1,2</sup>, XIA Cheng-Yi<sup>1,2</sup>

<sup>1</sup>(Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin 300384, China)

<sup>2</sup>(Engineering Research Center of Learning-Based Intelligent System (Ministry of Education), Tianjin University of Technology, Tianjin 300384, China)

**Abstract:** In large-scale, data-intensive application scenarios, the disadvantages of row storage access to data become increasingly prominent, which is gradually unable to meet the performance requirements for high-speed data access, and data needs more efficient transmission and processing methods. Therefore, it is of great significance to expand new memory access methods compatible with row and column access at the same time to improve access efficiency, reduce overall power consumption, and save memory space. This study focuses on dynamic random storage and non-volatile storage to introduce the column-oriented memory access method in detail, highlighting the analysis of the structural design for storage units and the implementation of the column-oriented memory access process. Finally, the two different ways of memory access are compared and summarized, and row- and column-oriented in-memory databases, data mining, data encryption algorithms, and the application scenarios of real-time systems are forecasted.

**Key words:** memory; storage structure; data access; non-volatility; column access

### 1 概述

随着人工智能<sup>[1]</sup>的兴起, 社会步入大数据时代, 数据成为密集应用的重要因素, 密集应用能够在短期内

产生大量的 TB 级 ( $2^{50}$  B) 甚至 PB 级 ( $2^{60}$  B) 数据, 它通过实时的数据处理和及时反馈, 动态调整应用的各个参数, 以此来优化应用性能. 它不仅可以通过分析同

① 基金项目: 国家自然科学基金 (61773286)

Foundation item: National Natural Science Foundation of China (61773286)

收稿时间: 2021-02-26; 修改时间: 2021-03-28; 采用时间: 2021-04-27

维度大量数据获取最佳的数据区间, 而且还能通过分析不同维度数据, 获取潜在的联系<sup>[2]</sup>, 例如沃尔玛的啤酒和纸尿裤摆放位置. 在处理海量的数据方面, 存在 CPU 处理性能发挥不充分、数据传输带宽制约问题<sup>[3]</sup>, 同时海量数据存储也面临巨大挑战<sup>[4]</sup>. 在截至 22 nm 工艺之前, 根据摩尔定律 CPU 每 18 个月性能翻一番. 与此同时, 内存的封装管脚数每 6 年会翻一番<sup>[5]</sup>. 在大多数应用中, 涉及到内存访问的指令占 20%–40%<sup>[6]</sup>. 因此, CPU 与内存之间存在着大量的数据传输. 然而, CPU 的执行速度和内存数据传输速度严重的不匹配<sup>[7]</sup>, 难以最大化发挥 CPU 的性能进而影响计算机的整体性能. 此外, 海量数据不同的存储方式也对访问的性能有着巨大的影响. 目前主流的内存访问方式都是面向行向的. 在特定大规模密集数据应用场景下, 行向数据访问方式已经不能够满足实时数据、快速访问、特定数据高效访问的需求, 同时行向的数据在访问过程中还存在着数据访问效率不高, 存储空间开销大的问题. 因此, 传统的内存访问面临着如下几个方面的挑战. 首先, 传统行方向的内存访问在获取数据的时候, 需要行方向和列方向分别选通. 其次, 在传统行方向的大量数据获取中, 为了匹配 CPU 的高性能数据都是通过组的形式并行传输的 (例如 64 bit), 这就导致了很多数据并不是真正需要的数据, 在筛选数据的阶段又会占用 CPU 时间影响性能. 最后, 在基于电荷材质组成的内存中, 由于受到摩尔定律的影响, 在单元尺寸已经进入了瓶颈, 无法再进一步的缩小尺寸——单元尺寸越小, 元器件间隙就越小, 产生的能耗越低, 性能越高<sup>[8]</sup>.

近年来, 列的内存访问方式应运而生. 新型存储单元的设计不仅能够实现列方向的内存访问而且还兼容行方向的内存访问, 这使得数据的访问方式更加的完善、高效. 此外, 也可以通过多层的方式来实现行列两种不同内存访问方式. 另外, 不同于电荷单元的电阻式单元也相继出现, 电阻式非易失性存储材料具有非易失、低能耗以及相较于电容式材料能够进一步缩小单元尺寸的特点<sup>[9]</sup>, 这些优秀的性能逐渐受到研究者的关注.

动态随机访问存储器<sup>[10]</sup>工作原理是利用电容内存储电荷的多寡来代表一个二进制比特是 1 还是 0. 面向行的内存访问方式效率已经非常的低下, 这成为影响内存访问效率的严重问题. 在继承传统的 DRAM 结构上, 通过设计新的存储单元组成新的存储阵列来实现

列向的访问方式. 在基于 DRAM 的基础结构上设计新的存储单元, 通过添加晶体管元器件使得 DRAM 不仅能够进行面向行的内存访问而且能够进行面向列的内存访问.

此外, 非易失性随机访问存储器 (Non-Volatile random access Memory, NVM)<sup>[11]</sup>有望可用于取代 DRAM 构建下一代内存体系存储容器. 尽管在众多材料之中存储单元相差各异, 然而在结构布局上可以有着相同的构建方式, 运用三维空间位置实现同等行向访问的列向访问方式.

内存的列向访问方式是应用于特定的情况下, 针对密集同类型数据的存储、访问、计算有着非常大的优势, 能够实现传统行向访问的数倍性能提升. 本文通过从内存列向访问出发, 详细的阐述了 DRAM、NVM 两个不同的实现方式以及对应的上层设计, 并对相关应用场景进行了展望.

## 2 DRAM 列向访问研究

### 2.1 DRAM 基础单元

在 DRAM 的存储结构设计中, 如图 1 所示存储单元是一个 1T1C 结构<sup>[12]</sup>, 其中 T 代表晶体管 (transistor), C 代表电容器 (capacitor), 通过电容中电荷量以及感知放大器 (Sense Amplifier, SA) 可以区分数据的逻辑 0 和 1.

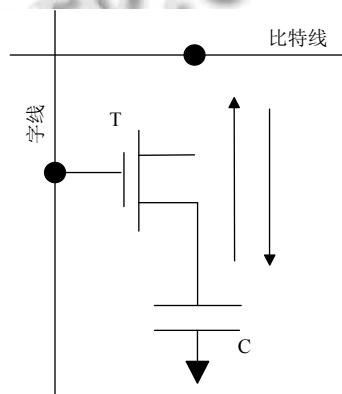


图 1 1T1C 结构示意图

### 2.2 不同的布局

由于传统的 DRAM 给每个单元安排一维的线性访问方式, 本文称之为面向行的访问. 在传统的 DRAM 中, 不同的数据布局方式可以有效地提高数据的缓存命中率. 并且可以通过行列转换来实现应用的列向访

问模式. 但是当转换矩阵过大时, 性能并不理想<sup>[13]</sup>. 文献 [14] 提出可以改变数据组织方式和访问模式, 使得数据获取性能提升了 1 倍. 另外, 文献 [15] 提出利用方块布局高效地收集块数据, 并将其运用于图形处理应用之中. 其次, Chatterjee 等人<sup>[16]</sup> 提出一种非线性的数据布局, 在该布局下实现了接近行布局方式下的性能<sup>[17]</sup>. 与此同时, Dahlgren 等人<sup>[18]</sup> 研究表明空间位置和访问模式是数据获取的重要因素.

因此, 考虑新的结构布局 and 空间位置能够有效提升数据的获取性能.

### 2.3 新型存储单元

在 1T1C 的结构构建的阵列中, 每次获取数据都是以行为单位进行的, 但是是一些多维数据分析的应用场景下, 该访问方式存在获取额外无效数据, 额外筛选数据操作的情况.

为此, Chen 等人<sup>[19]</sup> 创造性地设计出类似于面向行的数据访问方式, 本文称该为面向列的访问方式. 他们为此设计出了新的存储单元 2T1C. 如图 2 所示, 在访问一个共用电容结构中, 提供了两个不同方向的晶体管. 相比较 1T1C 而言, 该结构不同的设计在于存储单元上添加了一个用于控制列向访问的列晶体管, 通过控制行列晶体管开关状态, 就可以两种方式进行切换.

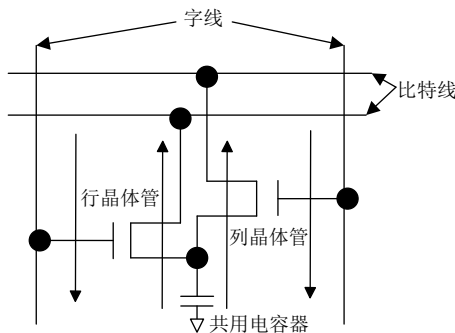


图 2 2T1C 结构示意图

在该结构下, Yu 等人<sup>[20]</sup> 研究表明: 相比较传统的 DRAM 结构, 在牺牲了 2.6% 的额外元器件开销代价后, 平均内存利用提升了 5.8%、内存访问延迟降低了 14.8%、能效消耗降低了 4.4%.

根据不同的存储单元, 最后结果对比结果如表 1 所示.

### 2.4 新型单元存储阵列

如图 3 中所示, 图 3 上半部分横向的虚线框中存储单元形成组称之为行, 图 3 右方纵向的虚线框中存

储单元形成组称之为列. 存储单元的晶体管控制的线组成字线, 传输电荷信息线组成比特线.

表 1 两种不同的 DRAM 的比较

| RAM    | 存储单元 | 行访问 | 列访问 |
|--------|------|-----|-----|
| 传统DRAM | 1T1C | √   | ×   |
| 新型DRAM | 2T1C | √   | √   |

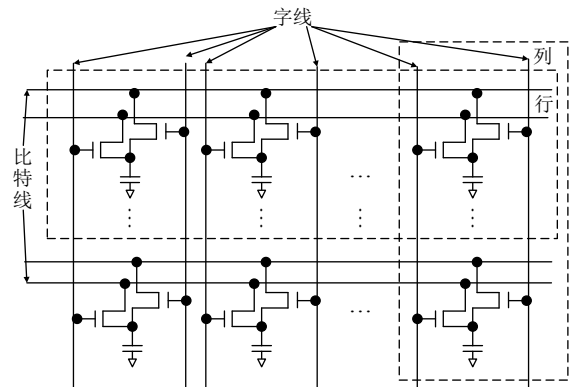


图 3 2T1C 单元存储阵列示意图

### 2.5 DRAM 内存访问

在基于 2T1C 存储单元的内存阵列结构中, 内存访问过程如图 4 中所示, CPU 先向内存控制器发去访问指令, 内存控制器通过请求, 传输的地址信息通过控制信号去进行判断行列访问类别, 然后传输地址信息选中对应的位置通过数据总线进行读写操作. 过程如图 4 所示.

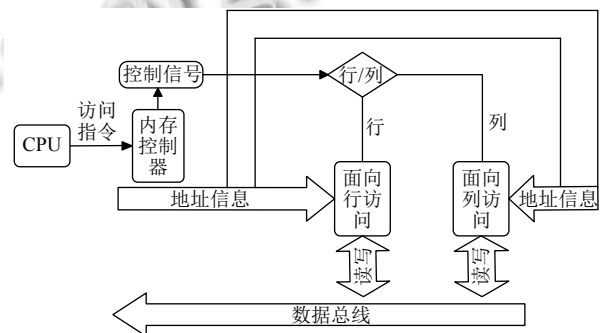


图 4 内存访问示意图

### 2.6 小结

考虑到行列转换过程中, 尽管可以利用行转换成列方式, 但是数据在转换过程中, 转换矩阵过大会导致转换效率不高, 从而数据访问的效率不佳. 另外, 改变数据的组织方式虽然可以提升数据的访问性能, 但是它结构布局没变, 仅是转换地址信息. 其次, 非线性的

布局方式能媲美行布局方式,有力的说明了非线性布局的有效性,因此行列布局也是得以研究,可以考虑不同的布局方式来实现已有的性能或者性能提升.最后,行列布局能够从结构出发,通过两种不同的路径访问同一存储单元,它不仅在数据获取性能上超过了行布局,并且能够应用于更多的场景下.

另外,非易失性材料有着许多材料性能优势,并且别出新意的结构排布,使得它有着超越前者的巨大研究潜力.

### 3 NVM的列向访问研究

NVM内存具有非易失、按字节存取、存储密度高、低能耗、读写性能接近DRAM、就地更新等众多优点<sup>[21]</sup>,得到了学术界和工业界的广泛研究<sup>[22]</sup>.大量的非易失性的存储器如雨后春笋一般出现,如电阻式随机存储器(ReRAM)<sup>[23]</sup>、自旋转移力矩磁随机存取存储器(STT-MRAM)<sup>[24]</sup>和相变内存(PCM)<sup>[25]</sup>以及忆阻器(memristor)<sup>[26]</sup>等.

NVM具有的优良特性已经应用于很多领域,包括数据库方面<sup>[27]</sup>和科学应用<sup>[28]</sup>等.

另外,文献[29]通过研究NVM构建异构内存系统,并且对其异构内存的页面管理进行研究以解决数据迁移的问题.通过内存分配策略,实现多维场景的优化.此外,文献[30]针对NVM的寿命研究,提出里内存压缩策略,尽量减少数据量,降低写入磨损,并且通过研究数据放置策略,提出基于轮转的磨损均衡机制,均匀使用资源,延长寿命.另外,文献[31]提出的混合内存存储模型除了拥有ReRAM的优秀特性之外还有着不错的读写性能.最后,ReRAM优秀的特点不仅可以运用于构建行存储结构,同样也可以运用构建与行列存储结构之中.

#### 3.1 ReRAM存储单元

ReRAM的读取延迟性能可以与DRAM相当,并且通过优化可以将DRAM和ReRAM之间的性能桥接在10%以内,被认为是构建下一代主存储器的替代方案<sup>[32]</sup>.因此,本文选择ReRAM来进行说明,以下使用NVM对ReRAM统称.

图5展示了ReRAM单元<sup>[33-35]</sup>的金属绝缘体金属结构:上电极、下电极和夹在它们之间的金属氧化物层<sup>[36]</sup>.通过施加外部电压,ReRAM单元可以在高阻状态(High Resistance State, HRS)和低电阻状态(Low

Resistance State, LRS)之间切换,这两个状态分别用来表示逻辑0和1<sup>[37]</sup>.

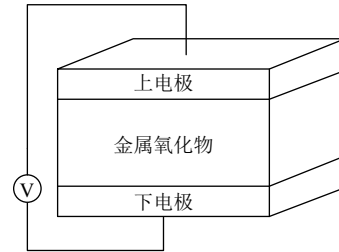


图5 ReRAM存储单元示意图

#### 3.2 交叉点结构

图6显示了由0T1R结构存储单元组成的阵列,其中T代表晶体管,R代表电阻器(resistor)<sup>[38,39]</sup>.直接通过上端和下端的电极连接上字线和比特线,不仅使存储单元的特征尺寸最小化,而且还具有良好的对称性,称为交叉点结构(crossbar)<sup>[40-42]</sup>.

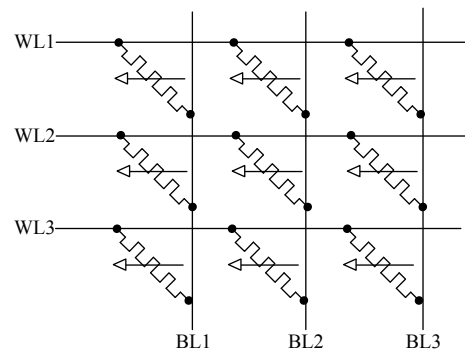


图6 交叉点结构示意图

图7是该结构的放大图<sup>[43]</sup>,在交叉点结构中存储单元仅是一个可以改阻值的电阻器,并且这个交叉点阵列是由众多的字线和比特线交叉组成的,每个存储单元位于字线和比特线的交点处属于一个三维的空间.从结构访问上看,由于行和列严格对称结构,所以在面向行和面向列的访问上有着得天独厚的结构优势.

#### 3.3 NVM内存访问

NVM的内存访问和DRAM类似,由于Crossbar结构,只需要获取对应的访问指令,通过相对应的控制指令去执行行列访问的不同的操作.

然而,在上述的交叉点结构中,会产生一个干扰问题,即潜入路径电流,它会引发相邻存储单元间的串扰干扰,从而导致误读,进而严重影响交叉点阵列的工作.



若能解决潜入路径电流问题,将会大大降低阵列的功耗,提高它的可靠性和稳定性.

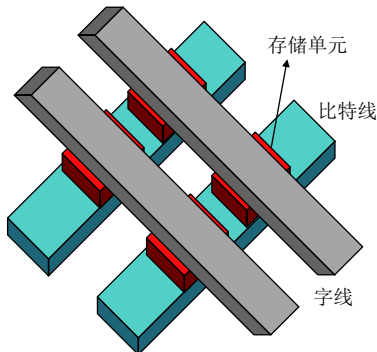


图7 交叉点结构放大图

### 3.4 2D-NVM 行列访问

为了解决潜入路径电流的问题,Chen 等人<sup>[44]</sup>提出了一种基于 ReRAM 的多层内存架构,该架构是由两层 ReRAM 存储单元和两个控制晶体管组成,可以实现行列的 2D 阵列的访问.如图 8 所示, M1 和 M2 存储单元, N1 和 N2 是两个控制晶体管. N1 控制行访问模式连接 iRB 和 iRW, N2 控制列访问模式连接 iCW 和 iCB. 通过测试结果显示:提出的设计结构上相对于交叉点结构提高了 5 倍左右的带宽,降低了 1.4 倍左右的能耗.

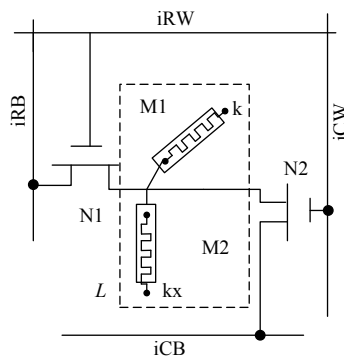


图8 存储单元结构示意图

### 3.5 小结

在非易失性存储中,实现行列访问的方式不同,表 2 中对两种不同的 NVM 进行了对比.

表2 NVM 的对比

| NVM类别  | 行访问 | 列访问 | 单元干扰 |
|--------|-----|-----|------|
| NVM    | √   | √   | √    |
| 2D-NVM | √   | √   | ×    |

表 3 显示了从物理设计结构上 DRAM 和 NVM 在实现列访问的不同设计以及相关的不同特点,并进行了比较和总结.

表3 行列访问实现的比较

| RAM      | 实现方式     | 特点      | 行 | 列 |
|----------|----------|---------|---|---|
| DRAM     | 无        | —       | √ | × |
| (新) DRAM | 添加元件     | 开销大     | √ | √ |
| NVM      | Crossbar | 简单,非易失性 | √ | √ |
| 2D-NVM   | 多层       | 结构复杂    | √ | √ |

考虑到上述的结构特点,尽管在 NVM 有着潜在电流的问题,但是 2D-NVM 添加一倍的器件成本使得在更通用的场景下不是最好的折中选择,使得基于 Crossbar 的 NVM 会是研究未来行列访问的最好选择.

通过添加或者修改物理结构之后,欲实现有效的行列访问还需要与其相对应的控制器设计、配套指令集、相应地址映射方式.

## 4 上层设计

由于 DRAM 的控制协议已经非常成熟了,而 DRAM 的控制协议不能够直接用于 NVM 之上.考虑到 DRAM 和 ReRAM 行列不兼容、读写刷新与否的技术特性存在的问题, Talati 等人<sup>[45]</sup>提出了 ReRAM 访问协议 R-DDR, 该协议的特点是通过列访问模式, 闭页策略 (closed-page polices). 并且通过测试 ReRAM 内存系统与使用 DDR4 在 SPEC CPU2006 工作负载访问下进行比较, 结果表明: 该系统的性能提高了 35%, 能耗降低了 7%.

在 DRAM 中针对内存访问拥有两个不同的访问指令, load 和 store. 然而, 为了更好的使用硬件资源, 需要设计对应的列访问的访问指令 Cstore 和 Cload<sup>[46]</sup>:

```
Cload reg, caddr;
Cstore reg, caddr;
```

reg 是目标寄存器, caddr 是面向列的数据地址. Cload 和 load 指令类似, Cstore 和 store 指令类似, 通过这两个新添加指令来配合内存控制器进行行列访问类型判断, 并且和存在的 load/store 指令搭配一起使用.

另外, 为了与行访问区分开来, 需要对列访问设计地址映射. 图 9 显示了面向行的 32 位地址结构, 图 10 显示了面向列的 32 位地址结构. 单位是 bit, Rank 是 2 个单位, Subarray 是 3 个单位, Bank 是 3 个单位, Channel 是 1 个单位, Row 和 Column 都是 10 个单位,

IntraBus 是 3 个单位. 行地址结构和列的地址结构基本类似, 唯一不同的点在于, 两者之间 Row 和 Column 顺序不同, 就由此来进行区分两种不同的地址.

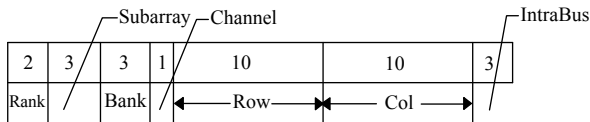


图9 行地址映射

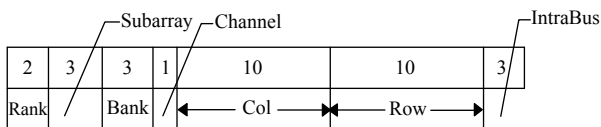


图10 列地址映射

## 5 总结与展望

本文从内存访问方式的角度进行研究, 分析了传统行向访问过程中存在的特定场景性能不足的问题, 为了满足特定的性能需求拓展新的列向内存访问方式, 具体总结如下:

在 DRAM 设计中, 由于存储单元上添加了一个列向的晶体管, 使得存储单元可以列向数据访问. 但是, 添加晶体管导致存储单元的尺寸变大, 不仅增加了格外的开销, 还会限制了性能进一步的提升. 考虑到 DRAM 需要定时刷新, 两个晶体管共同访问同一个电容器, 因此, 刷新次数也会相应的增加, 同时会增加刷新能耗.

在 NVM 设计中, 基于 Crossbar 的三维叠加结构, 使得行列访问方式可以得到统一. 得益于这个结构, 非易失性材料能够满足存储单元尺寸的更小化, 同时还能降低访问的能耗. 但是, 这种结构需要严格的对称性, 访问还会对邻居单元的造成干扰. 因此, 2D-NVM 的设计就是在降低了存储单元之间干扰的基础上, 添加了多层的结构区分数据访问的不同方式. 然而, 尽管 2D-NVM 的设计解决了上述问题, 但是带来的是更加复杂的存储单元和更大单元开销和能耗.

针对列向访问的应用场景, 可在以下 4 个方面进一步丰富其应用:

(1) 根据内存计算<sup>[47]</sup>概念, 结合内存数据库 (In-Memory DataBase, IMDB)<sup>[27]</sup>的应用, 使用列向访问方式获取列向数据库<sup>[48]</sup>的列向数据, 可以很方便获取所需的特定数据, 还能够同时对不同列的数据可以聚合

和分类<sup>[49]</sup>.

(2) 当列向访问应用于大数据的挖掘<sup>[50]</sup>中, 考虑到存储的数据基于相同的数据类型或者数据结构, 同类型数据的获取将会更加的高效, 并且对于数据之间的关系分析, 将会更加的准确.

(3) 当行列访问应用于数据加密算法<sup>[51]</sup>中, 考虑到加密算法需要对已有数据进行大量的数学运算, 其中计算较多的是矩阵相乘, 通过列向访问可以快速的获取矩阵中存储的一行或者一列数据, 然后进行大量的矩阵相乘运算得到最终的结果. 列向的访问不仅能够缩短数据获取的时间, 还能提高加密算法的时效性.

(4) 当列向访问结合实时微体系结构<sup>[52]</sup>应用于实时处理时, 通过列向访问数据并对数据进行快速处理, 能够给予其更加快速的实时反馈, 例如存储汽车行驶中的各项参数, 通过数据实时计算能够快速调整无人驾驶汽车的速度或方向.

当前, 传统的行向访问的内存获取方式应用于通用的场景. 然而, 内存的列向访问是针对某种特定的场景设计的优化数据获取方式, 尤其是密集数据的存储. 由于数据行列访问方式在一个具体的应用中有着较大的性能差距, 所以列向访问会极大的提升特定应用场景下的访问效率. 因此, 后续会对内存数据库的数据存储和访问性能提升进行深入的研究.

## 参考文献

- Bringsjord S, Schimanski B. What is artificial intelligence? Psychometric AI as an answer. Proceedings of the 18th International Joint Conference on Artificial Intelligence. Acapulco: ACM, 2003. 887–893.
- George G, Haas MR, Pentland A. Big data and management. Academy of Management Journal, 2014, 57(2): 321–326. [doi: 10.5465/amj.2014.4002]
- Sun WQ, Li FQ, Guo W, et al. Store, schedule and switch — A new data delivery model in the big data era. 2013 15th International Conference on Transparent Optical Networks (ICTON). Cartagena: IEEE, 2013. 1–4.
- Goli-Malekabadi Z, Sargolzaei-Javan M, Akbari MK. An effective model for store and retrieve big health data in cloud computing. Computer Methods and Programs in Biomedicine, 2016, 132: 75–82. [doi: 10.1016/j.cmpb.2016.04.016]
- Stanley-Marbell P, Cabezas VC, Luijten R. Pinned to the walls: Impact of packaging and application properties on the

- memory and power walls. Proceedings of the 17th IEEE/ACM International Symposium on Low-Power Electronics and Design. Fukuoka: IEEE, 2011. 51–56.
- 6 Hennessy JL, Patterson DA. Computer Architecture: A Quantitative Approach. 4th ed. San Mateo: Morgan Kaufmann, 2006. 599–600.
- 7 Wulf WA, Mckee SA. Hitting the memory wall: Implications of the obvious. SIGARCH Computer Architecture News, 1995, 23(1): 20–24. [doi: [10.1145/216585.216588](https://doi.org/10.1145/216585.216588)]
- 8 Ramkumar K, Kouznetsov I, Prabhakar V, *et al.* A scalable, low voltage, low cost SONOS memory technology for embedded NVM applications. 2013 5th IEEE International Memory Workshop. Monterey: IEEE, 2013. 199–202.
- 9 Jagasivamani M, Walden C, Singh D, *et al.* Design for ReRAM-based main-memory architectures. Proceedings of the International Symposium on Memory Systems. Washington: ACM, 2019. 342–350.
- 10 Scheick LZ, Guertin SM, Swift GM. Analysis of radiation effects on individual DRAM cells. IEEE Transactions on Nuclear Science, 2000, 47(6): 2534–2538. [doi: [10.1109/23.903804](https://doi.org/10.1109/23.903804)]
- 11 李月, 王芳. 基于 NVM 的存储安全综述. 计算机科学, 2018, 45(7): 53–60.
- 12 Smith FJ, Yu RT, Lee I, *et al.* A 64 Kbit MOS dynamic RAM with novel memory capacitor. IEEE Journal of Solid-State Circuits, 1980, 15(2): 184–189. [doi: [10.1109/JSSC.1980.1051361](https://doi.org/10.1109/JSSC.1980.1051361)]
- 13 Chatterjee S, Sen S. Cache-efficient matrix transposition. Proceedings 6th International Symposium on High-Performance Computer Architecture. HPCA-6 (Cat. No. PR00550). Toulouse: IEEE, 2000. 195–205.
- 14 Seshadri V, Mullins T, Boroumand A, *et al.* Gather-scatter DRAM: In DRAM-address translation to improve the spatial locality of non-unit strided accesses. Proceedings of the 48th International Symposium on Microarchitecture. Waikiki: ACM, 2015. 267–280.
- 15 Pharr M. GPU Gems 2: Programming Techniques for High-performance Graphics and General-Purpose Computation. Upper Saddle River: Addison-Wesley Professional, 2005.
- 16 Chatterjee S, Jain VV, Lebeck RA, *et al.* Nonlinear array layouts for hierarchical memory systems. Proceedings of the 13th International Conference on Supercomputing. Rhodes: ACM, 1999. 444–453.
- 17 Thiyagalingam J, Beckmann O, Kelly PHJ. An exhaustive evaluation of row-major, column-major and morton layouts for large two-dimensional arrays. Proceedings of the 19th Annual UK Performance Engineering Workshop. 2003. 1–12.
- 18 Dahlgren F, Stenstrom P. Evaluation of hardware-based stride and sequential prefetching in shared-memory multiprocessors. IEEE Transactions on Parallel and Distributed Systems, 1996, 7(4): 385–398. [doi: [10.1109/71.494633](https://doi.org/10.1109/71.494633)]
- 19 Chen YH, Liu YY. Dual-addressing memory architecture for two-dimensional memory access patterns. Proceedings of the Conference on Design, Automation and Test in Europe 2013. Grenoble: ACM, 2013. 71–76.
- 20 Yu J, Jang W. FDRAM: DRAM architecture flexible in successive row and column accesses. 2015 33rd IEEE International Conference on Computer Design (ICCD). New York: IEEE, 2015. 451–454.
- 21 Kim K, Koh GH. Future memory technology including emerging new memories. 2004 24th International Conference on Microelectronics (IEEE Cat. No. 04TH8716). Nis: IEEE, 2004. 377–384.
- 22 肖仁智, 冯丹, 胡燊翀, 等. 面向非易失内存的数据一致性研究综述. 计算机研究与发展, 2020, 57(1): 85–101. [doi: [10.7544/issn1000-1239.2020.20190062](https://doi.org/10.7544/issn1000-1239.2020.20190062)]
- 23 Mao MQ, Cao Y, Yu SM, *et al.* Optimizing latency, energy, and reliability of 1T1R ReRAM through appropriate voltage settings. 2015 33rd IEEE International Conference on Computer Design (ICCD). New York: IEEE, 2015. 359–366.
- 24 Kültürsay E, Kandemir M, Sivasubramaniam A, *et al.* Evaluating STT-RAM as an energy-efficient main memory alternative. 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). Austin: IEEE, 2013. 256–267.
- 25 张鸿斌, 范捷, 舒继武, 等. 基于相变存储器的存储系统与技术综述. 计算机研究与发展, 2014, 51(8): 1647–1662. [doi: [10.7544/issn1000-1239.2014.20131123](https://doi.org/10.7544/issn1000-1239.2014.20131123)]
- 26 Strukov DB, Snider GS, Stewart DR, *et al.* The missing memristor found. Nature, 2008, 453(7191): 80–83. [doi: [10.1038/nature06932](https://doi.org/10.1038/nature06932)]
- 27 Arulraj J, Pavlo A. How to build a non-volatile memory database management system. Proceedings of the 2017 ACM International Conference on Management of Data. Illinois: ACM, 2017. 1753–1758.
- 28 Li D, Vetter SJ, Marin G, *et al.* Identifying opportunities for byte-addressable non-volatile memory in extreme-scale scientific applications. 2012 IEEE 26th International Parallel and Distributed Processing Symposium. Shanghai: IEEE, 2012. 945–956.

- 29 陈吉强. 异构内存系统中页面管理的优化设计 [硕士学位论文]. 合肥: 中国科学技术大学, 2019.
- 30 叶源远. 面向 3D 堆叠的非易失性内存压缩和磨损均衡技术 [硕士学位论文]. 武汉: 华中科技大学, 2019.
- 31 杜娇. 基于阻变存储器的混合存储模型研究 [硕士学位论文]. 乌鲁木齐: 新疆大学, 2018.
- 32 Xu C, Niu DM, Muralimanohar N, *et al.* Overcoming the challenges of crossbar resistive memory architectures. 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA). Burlingame: IEEE, 2015. 476–488.
- 33 Xu C. Modeling, circuit design, and microarchitectural optimization of emerging resistive memory [Ph.D. Thesis]. State College: The Pennsylvania State University, 2014.
- 34 Chi P, Li SC, Xu C, *et al.* PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. ACM SIGARCH Computer Architecture News, 2016, 44(3): 27–39. [doi: [10.1145/3007787.3001140](https://doi.org/10.1145/3007787.3001140)]
- 35 Liu CC, Wu T. ReRAM-based circuit and system design for future storage and computing. 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). Chengdu: IEEE, 2018. 390–393.
- 36 Wong HSP, Lee HY, Yu SM, *et al.* Metal-oxide RRAM. Proceedings of the IEEE, 2012, 100(6): 1951–1970. [doi: [10.1109/JPROC.2012.2190369](https://doi.org/10.1109/JPROC.2012.2190369)]
- 37 Xu C, Niu DM, Muralimanohar N, *et al.* Understanding the trade-offs in multi-level cell ReRAM memory design. 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC). Austin: IEEE, 2013. 1–6.
- 38 Kau D, Tang S, Karpov IV, *et al.* A stackable cross point phase change memory. 2009 IEEE International Electron Devices Meeting (IEDM). Baltimore: IEEE, 2009. 1–4.
- 39 Lee HD, Kim SG, Cho K, *et al.* Integration of 4F2 selector-less crossbar array 2 Mb ReRAM based on transition metal oxides for high density memory applications. 2012 Symposium on VLSI Technology (VLSIT). Honolulu: IEEE, 2012. 151–152.
- 40 Talati N, Gupta S, Mane P, *et al.* Logic design within memristive memories using Memristor-Aided loGIC (MAGIC). IEEE Transactions on Nanotechnology, 2016, 15(4): 635–650. [doi: [10.1109/TNANO.2016.2570248](https://doi.org/10.1109/TNANO.2016.2570248)]
- 41 Chen A. A review of emerging Non-Volatile Memory (NVM) technologies and applications. Solid-State Electronics, 2016, 125: 25–38. [doi: [10.1016/j.sse.2016.07.006](https://doi.org/10.1016/j.sse.2016.07.006)]
- 42 Niu DM, Xu C, Muralimanohar N, *et al.* Design trade-offs for high density cross-point resistive memory. Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design. Redondo Beach: ACM, 2012. 209–214.
- 43 Wang P, Li S, Sun GY, *et al.* RC-NVM: Enabling symmetric row and column memory accesses for in-memory databases. 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA). Vienna: IEEE, 2018. 518–530.
- 44 Chen Y, Zhang J, Xu YB, *et al.* A ReRAM-based row-column-oriented memory architecture for convolutional neural networks. IEICE Transactions on Electronics, 2019, E102. C(7): 580–584. [doi: [10.1587/transele.2018CTS0001](https://doi.org/10.1587/transele.2018CTS0001)]
- 45 Talati N, Ha H, Perach B, *et al.* CONCEPT: A column-oriented memory controller for efficient memory and PIM operations in RRAM. IEEE Micro, 2019, 39(1): 33–43. [doi: [10.1109/MM.2018.2890033](https://doi.org/10.1109/MM.2018.2890033)]
- 46 Li S, Xiao N, Wang P, *et al.* RC-NVM: Dual-addressing non-volatile memory architecture supporting both row and column memory accesses. IEEE Transactions on Computers, 2019, 68(2): 239–254. [doi: [10.1109/TC.2018.2868368](https://doi.org/10.1109/TC.2018.2868368)]
- 47 罗乐, 刘轶, 钱德沛. 内存计算技术研究综述. 软件学报, 2016, 27(8): 2147–2167. [doi: [10.13328/j.cnki.jos.005103](https://doi.org/10.13328/j.cnki.jos.005103)]
- 48 Stonebraker M, Abadi DJ, Batkin A, *et al.* C-store: A column-oriented DBMS. Proceedings of the 31st International Conference on Very Large Data Bases. Trondheim: ACM, 2005. 553–564.
- 49 Abadi DJ, Boncz PA, Harizopoulos S. Column-oriented database systems. Proceedings of the VLDB Endowment, 2009, 2(2): 1664–1665. [doi: [10.14778/1687553.1687625](https://doi.org/10.14778/1687553.1687625)]
- 50 杨小娟. 数据挖掘国内研究综述. 电脑编程技巧与维护, 2020, (8): 115–117. [doi: [10.3969/j.issn.1006-4052.2020.08.042](https://doi.org/10.3969/j.issn.1006-4052.2020.08.042)]
- 51 于琳, 王龙葛. 基于矩阵乘法的加密算法研究. 信息系统工程, 2016, (4): 136, 138.
- 52 石伟, 张明, 郭御风, 等. 实时微处理器体系结构综述. 计算机工程与科学, 2015, 37(5): 857–864. [doi: [10.3969/j.issn.1007-130X.2015.05.002](https://doi.org/10.3969/j.issn.1007-130X.2015.05.002)]