**L13**

**Điều phối dữ liệu (Data Orchestration)** là một khái niệm quan trọng trong quá trình ánh xạ các phép tính của Mạng Nơ-ron Sâu (DNN) lên phần cứng, tập trung vào việc quản lý di chuyển và lưu trữ dữ liệu nhằm tối ưu hiệu suất. Các nguyên tắc chính trong điều phối dữ liệu bao gồm:

* **Tái sử dụng hiệu quả (Efficient Reuse):** Sử dụng bộ nhớ nhỏ được đặt gần các đơn vị tính toán để dễ dàng tái sử dụng dữ liệu, giảm thời gian truy xuất.
* **Đồng bộ chính xác (Precise Synchronization):** Chỉ chờ đúng dữ liệu cần thiết và phản hồi nhanh chóng, tránh việc chờ đợi không cần thiết hoặc kiểm tra dữ liệu từ xa.
* **Chồng chéo giữa tải và sử dụng dữ liệu (Delivery/Use Overlap):** Đảm bảo ô dữ liệu tiếp theo có sẵn ngay khi ô hiện tại được xử lý xong, thường được thực hiện thông qua bộ đệm kép (*double-buffering*).
* **Hiệu quả sử dụng bộ nhớ (Storage Usage Efficiency):** Giảm thiểu bộ nhớ nhàn rỗi bằng cách tránh độ trễ truy xuất dữ liệu kéo dài.
* **Hiệu suất băng thông (Bandwidth Efficiency):** Tối đa hóa tốc độ truyền dữ liệu bằng cách kiểm soát số lượng yêu cầu đang xử lý cùng lúc.
* **Chia sẻ dữ liệu giữa các đơn vị tính toán (Cross-Unit Use):** Giảm chi phí truy cập dữ liệu và truyền thông bằng cách chia sẻ dữ liệu giữa nhiều đơn vị tính toán khác nhau.

Các nguyên tắc này đóng vai trò quan trọng trong việc:

* Tối ưu hóa hiệu suất của hệ thống
* Giảm thiểu độ trễ
* Tiết kiệm năng lượng
* Sử dụng hiệu quả tài nguyên phần cứng
* Đảm bảo luồng dữ liệu trơn tru và hiệu quả

**Các Phương Pháp Điều Phối Dữ Liệu**

Có nhiều cách tiếp cận khác nhau, bao gồm:

* **Ẩn (Implicit) vs. Tường minh (Explicit):**
  + Phương pháp *implicit* (ẩn) tự động điều phối dữ liệu mà không cần sự kiểm soát trực tiếp từ lập trình viên.
  + Phương pháp *explicit* (tường minh) yêu cầu lập trình viên xác định rõ cách dữ liệu được di chuyển và lưu trữ.
* **Kết hợp (Coupled) vs. Tách rời (Decoupled):**
  + Phương pháp *coupled* gắn chặt việc tính toán và truy xuất dữ liệu, giúp đơn giản hóa nhưng có thể gây nghẽn dữ liệu.
  + Phương pháp *decoupled* tách rời truy xuất dữ liệu khỏi tính toán, tăng tính linh hoạt và hiệu suất.
* **Explicit Decoupled Data Orchestration**: Kết hợp điều phối dữ liệu rõ ràng và tách rời để tối ưu hóa hiệu suất. Đây được xem là cách tiếp cận ưu việt nhất.

A screenshot of a computer

AI-generated content may be incorrect.

**EDDO Strategies**

Một trong những chiến lược EDDO phổ biến là **Buffets**.

**Buffets là gì?**

* Buffets là một mô hình lưu trữ có thể mở rộng, giúp truy cập dữ liệu ngẫu nhiên trong một vùng nhớ đang hoạt động (*live window*).
* Cho phép cập nhật giá trị trong vùng nhớ mà không cần tải lại toàn bộ dữ liệu.
* So với các mô hình lưu trữ khác:
  + So với **FIFO**, buffets cho phép truy cập ngẫu nhiên thay vì chỉ theo thứ tự vào-ra.
  + So với **Scratchpad**, buffets bổ sung cơ chế kiểm soát đồng bộ (*scoreboarding*) giúp theo dõi trạng thái dữ liệu.
  + So với **Cache**, buffets giảm thiểu yêu cầu lưu trữ thông tin thẻ (*tag store*) và sử dụng mô hình *push* thay vì *pull* để tối ưu hóa băng thông.

**Các Đặc Tính Hành Vi của Buffet (Buffet Behavioral Attributes)**

Hành vi của buffet được xác định dựa trên trình tự địa chỉ liên quan đến việc **nạp dữ liệu (fill), đọc dữ liệu (read), và cập nhật dữ liệu (update)**.

* **Trình tự Nạp Dữ liệu (*Fill Sequence*)**
  + Dựa trên chuỗi địa chỉ "fill", buffet sẽ lấy dữ liệu từ kênh nạp cho đến khi bộ đệm đầy.
* **Trình tự Đọc Dữ liệu (*Read Sequence*)**
  + Dựa trên chuỗi địa chỉ "read", buffet sẽ đẩy dữ liệu xuống kênh đọc (*read LI channel*), nhưng chỉ khi giá trị đó đã được nạp (*filled*).
  + Chuỗi địa chỉ "read" cũng có thể thông báo cho buffet rằng một giá trị có thể bị loại bỏ để giải phóng không gian lưu trữ. Việc này được thực hiện thông qua cổng điều khiển thu nhỏ (*shrink control port*).
* **Trình tự Cập Nhật Dữ liệu (*Update Sequence*)**
  + Dựa trên chuỗi địa chỉ "update", buffet sẽ lấy giá trị từ kênh cập nhật (*update channel*).
* **Hỗ trợ nhiều bộ đệm logic bên trong một bộ đệm vật lý**
  + Một bộ đệm vật lý có thể chứa nhiều bộ đệm logic riêng biệt, giúp tối ưu hóa việc lưu trữ và xử lý dữ liệu.

A diagram of a window

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**Tóm tắt nội dung chính: Mapping Output Stationary (OS) to Hardware**

**Output Stationary (OS)** là một phương pháp ánh xạ dữ liệu lên phần cứng, giữ các giá trị đầu ra cố định trong **Processing Elements (PEs)** để giảm thiểu việc đọc/ghi dữ liệu, từ đó tối ưu hóa hiệu suất và năng lượng.

**🔍 Các yếu tố chính khi ánh xạ OS lên phần cứng**

**Tích chập 1D (1-D Convolution)**

* Đầu ra được tính bằng cách nhân các trọng số (**weights**) với các đầu vào tương ứng và cộng lại: Oq=Iq+s×FsO\_q = I\_{q+s} \times F\_sOq​=Iq+s​×Fs​
* Giảm thiểu truy xuất bộ nhớ ngoài nhờ tái sử dụng dữ liệu nhiều lần.

**Luồng dữ liệu trong một PE (Single PE Output Stationary Flow)**

* Dữ liệu và trọng số được truyền qua PE, trong khi tổng tạm thời (**partial sums**) được tích lũy trực tiếp tại PE.
* Không cần tính toán địa chỉ tải/lưu dữ liệu → Tăng hiệu suất xử lý.
* Tối ưu hóa việc đọc/ghi dữ liệu theo mô hình đồng bộ chu kỳ (**systolic**).

**Mô hình tham chiếu (Reference Pattern)**

* **Đầu ra** được sử dụng nhiều lần.
* **Trọng số** được sử dụng lại liên tục.
* Sử dụng **cửa sổ trượt (sliding window)** để tối ưu hóa việc lưu trữ dữ liệu đầu vào.

**Truy cập dữ liệu cấp L1 (L1 Data Accesses)**

* Số lần đọc và ghi dữ liệu:
  + **Phép tính nhân-tích lũy (MACs):** Q × S.
  + **Đọc trọng số (Weights):** Q × S.
  + **Đọc đầu vào (Inputs):** Q × S.
  + **Ghi đầu ra (Outputs):** Q.
  + Không có lần đọc đầu ra.

**Bộ nhớ đệm trung gian (Intermediate Buffering)**

* Giảm số lần truy xuất dữ liệu bằng cách thêm các cấp bộ nhớ đệm giữa bộ nhớ chính và PE.
* Tạo thêm các vòng lặp mới trong cấu trúc vòng lặp (**loop nest**).

**Ánh xạ không gian (Spatial Mapping)**

* Sử dụng nhiều PE để xử lý song song, tăng tốc độ tính toán.
* Cần hỗ trợ **giao tiếp giữa các PE (inter-PE communication)** và tổng hợp không gian (**spatial sum**).

**Trọng số cố định trong không gian (Spatial Weight Stationary - Skewed)**

* Giữ nguyên trọng số trong PE trong một khoảng thời gian dài để tối ưu hóa việc tái sử dụng dữ liệu.
* Cho phép truyền dữ liệu đa hướng (**multicast**) và giao tiếp hiệu quả giữa các PE.

**Các bước chính của quá trình ánh xạ**

1. **Chọn luồng dữ liệu (Dataflow Selection)**
   * Xác định **thứ tự vòng lặp** trong quá trình tính toán.
   * Điều này giúp kiểm soát luồng dữ liệu và cách truy xuất dữ liệu trong bộ nhớ.
2. **Chia nhỏ dữ liệu (Partitioning)**
   * Xác định **giới hạn vòng lặp** về không gian và thời gian.
   * Chia nhỏ dữ liệu và phép tính để xử lý song song trên nhiều phần tử xử lý (**PEs**).
3. **Chi tiết hành vi (Behavioral Details)**
   * Lựa chọn các chi tiết triển khai cụ thể như **bỏ qua một số cấp bộ nhớ** nếu không cần thiết để tăng hiệu suất.
4. **Gắn kết phép tính (Computation Binding)**
   * Gán các phép tính cụ thể cho từng phần tử phần cứng phù hợp.

**Mô hình tính toán CPU vs. DNN (CPU vs. DNN Compute Models)**

* **Mô hình CPU (CPU Compute Model)**
  + Bắt đầu từ bộ xử lý và trình biên dịch.
  + Xử lý dữ liệu đầu vào theo chương trình nhị phân.
  + Tạo dữ liệu đã xử lý và thống kê hành vi.
* **Mô hình DNN (DNN Compute Model)**
  + Bắt đầu với **đặc tả bài toán (problem specification)** và **hình dạng dữ liệu (shape)**.
  + Cấu hình **luồng dữ liệu** và các ràng buộc (**constraints**).
  + Thực thi mạng nơ-ron sâu (DNN) và ánh xạ các đầu vào, trọng số để tạo ra đầu ra tương ứng.

**Công cụ hỗ trợ ánh xạ (Mapping Tools)**

* **Timeloop**
  + Dùng để đánh giá và khám phá không gian thiết kế kiến trúc cho các bộ gia tốc DNN.
  + Hỗ trợ nhiều mô hình DNN khác nhau và tương thích với nhiều kiến trúc phần cứng.
* **Accelergy**
  + Công cụ ước lượng mức tiêu thụ năng lượng dựa trên mô tả kiến trúc và cách triển khai.

**Các đặc tả cần thiết cho ánh xạ (Specifications for Mapping)**

1. **Đặc tả khối lượng công việc (Workload Specification)**
   * Cung cấp thông tin về **không gian phép toán** và **không gian dữ liệu** cho công cụ Timeloop.
2. **Đặc tả kiến trúc (Architecture Specifications)**
   * Xác định các ánh xạ hợp lệ, thứ tự vòng lặp (**dataflows**) và các giới hạn vòng lặp liên quan.
   * Bao gồm các tính năng như **tái sử dụng theo thời gian (temporal reuse)**, **tái sử dụng không gian (spatial reuse)** và các ràng buộc như hạn chế về thứ tự chỉ số.
3. **Đặc tả triển khai (Implementation Specifications)**
   * Xác định **độ trễ (latency)** và **mức tiêu thụ năng lượng** của ánh xạ.
   * Bao gồm thông tin về băng thông bộ đệm, tổ chức các cổng bộ đệm, khả năng xử lý song song của các PE và độ trễ của mạng liên kết.