

平行處理

Flynn 分類 / SMP / 快取一致性與 MESI / 叢集 / NUMA / 多核心與 GPU

摘要

單一處理器的效能終有極限, 平行化是必然的出路。本章依 Flynn 分類法建立平行結構的地圖; 深入對稱多處理器 (SMP) 的組織與其核心難題——快取一致性, 詳述 MESI 協定; 接著介紹叢集 (cluster) 與 NUMA 兩種擴展路徑; 最後補充多核心時代的現實: 記憶體一致性模型、GPU 與資料平行。本章對應 Stallings 第 18 章 (第 6 版), 並更新至現代。

目錄

1	Flynn 分類法	2
2	對稱多處理器 (SMP)	2
3	快取一致性與 MESI	2
3.1	問題	2
3.2	MESI 協定	2
4	叢集與 NUMA	3
4.1	叢集	3
4.2	NUMA	3
5	多核心時代的補充	3
6	本章重點回顧	3
7	複習問題	4

1 Flynn 分類法

依指令流與資料流的多寡:

類別	全名	例子
SISD	單指令單資料	傳統單處理器
SIMD	單指令多資料	向量處理器、GPU 的 SIMT、RISC-V V 擴展
MISD	多指令單資料	(幾無實例)
MIMD	多指令多資料	SMP、叢集、NUMA、多核心

MIMD 再依記憶體組織細分: 共享記憶體 (SMP、NUMA) 與分散式記憶體 (叢集)。

2 對稱多處理器 (SMP)

SMP: 多個能力相同的處理器, 共享同一主記憶體與 I/O, 由單一作業系統統一管理, 任何處理器都能執行任何工作 (對稱)。優點: 效能 (工作可真平行)、可用性 (單一處理器失效不停機)、漸進擴充。

組織上最常見的是共享匯流排: 簡單、易擴充, 但匯流排是瓶頸——每顆處理器配備本地快取以減少匯流排流量, 於是引出一致性問題。

3 快取一致性與 MESI

3.1 問題

同一資料在多個快取中各有副本; 任一處理器更新自己的副本後, 其他快取中的舊值就過期了。軟體解法 (標記不可快取、編譯器分析) 過於保守; 主流是硬體協定:

- 目錄協定: 集中目錄記錄每個區塊在哪些快取——適合大規模、非匯流排互連 (NUMA)。
- 窺探協定 (snoopy): 每個快取控制器監聽匯流排上所有交易——適合共享匯流排; 代表即 MESI。

3.2 MESI 協定

寫回快取的每一行處於四態之一:

態	名稱	意義
M	Modified	本快取獨有且已修改; 與主記憶體不一致, 負責供應資料
E	Exclusive	本快取獨有、乾淨; 可直接改寫 (轉 M) 而無需匯流排交易
S	Shared	多個快取可能都有、乾淨; 寫入前須廣播無效化
I	Invalid	無效

關鍵轉移: 讀未命中 → 別的快取有 M 則由它供資料並轉 S (或寫回); 寫命中 S → 先發 **invalidate** 廣播, 他快取全轉 I, 自己轉 M; 窺探到他人讀取自己 M 的行 → 供資料、轉 S。

偽共享 false sharing

兩顆核心分別頻繁寫同一快取行內的不同變數, MESI 使該行在兩快取間反覆無效化——效能崩落。多執行緒程式把「各核心私有的熱資料」隔開到不同快取行 (padding), 即源於此。

4 叢集與 NUMA

4.1 叢集

一組完整的計算機 (節點) 以高速網路互連, 作為單一資源使用: 絕對可擴充 (數千節點)、高可用 (節點失效可接手)、性價比 (商用零件堆疊)。代價: 程式設計模型是訊息傳遞 (MPI), 資料須顯式分割搬移。超級電腦 TOP500 幾乎全是叢集; 雲端資料中心是其商業極致。

4.2 NUMA

介於 SMP 與叢集之間: 全系統單一位址空間, 但記憶體實體分散於各節點——存取本地記憶體快、遠端慢 (Non-Uniform)。CC-NUMA 以目錄協定維持全域快取一致。優點: 保有共享記憶體程式模型, 又能擴充到匯流排 SMP 無法到達的規模。現代雙插槽伺服器即是小型 NUMA; OS 的 NUMA-aware 配置 (記憶體就近分配) 直接影響效能。

5 多核心時代的補充

- 多核心 = 單晶片 **SMP**: 共享 L3、私有 L1/L2, MESI(F/O 變體) 依然是骨幹。
- 記憶體一致性模型: 一致性協定保證「單一位置」的秩序; 一致性模型約束「不同位置」寫入的可見順序。x86 提供強序 (TSO); **RISC-V** 採弱序 **RVWMO**, 需要順序時用 fence 指令。多執行緒無鎖程式必須理解這一層。
- **GPU**: 放棄 ILP 與快取容量, 以數萬執行緒隱藏延遲, SIMT 執行——資料平行工作 (圖形、深度學習) 的王者; 與 CPU 的 MIMD 互補。
- **RISC-V** 的平行支援: A 擴展 (原子操作 lr/sc、amoadd 等)、V 向量擴展、hart (硬體執行緒) 概念——在開放 ISA 上, 從微控制器到千核加速器共用同一套基礎。

6 本章重點回顧

本章要點

- Flynn: SISD/SIMD/MISD/MIMD; MIMD 分共享記憶體 (SMP/NUMA) 與訊息傳遞 (叢集)。
- SMP 共享匯流排 + 私有快取 → 一致性問題 → 窺探式 MESI 四態協定。
- 叢集以訊息傳遞換取無限擴充; NUMA 以非均勻延遲換取單一位址空間的擴充。
- 多核心 = 晶片上的 SMP; 弱記憶體序 (RVWMO) 與 fence 是無鎖程式的必修課。

7 複習問題

1. 以 Flynn 分類法歸類: 四核 x86、GPU、MPI 叢集、RISC-V 向量機。
2. MESI 中, E 態存在的價值是什麼 (相較只有 MSI)?
3. 描述: P1 讀 X (未命中) → P2 讀 X → P1 寫 X → P2 讀 X 過程中兩快取中 X 行的狀態變化。
4. 什麼是偽共享? 如何在程式層面避免?
5. SMP、NUMA、叢集在「位址空間、一致性、擴充性」三軸上的定位各為何?
6. RISC-V 的 fence 指令解決什麼問題? 為何 x86 程式移植到 RISC-V 可能暴露新的並行錯誤?

參考資料

- [1] W. Stallings, *Computer Organization and Architecture*, 6th ed., Chapter 18.
- [2] D. Sorin, M. Hill, D. Wood, *A Primer on Memory Consistency and Cache Coherence*, 2nd ed., Morgan & Claypool, 2020.
- [3] RISC-V International, *RVWMO Memory Consistency Model*, ISA Manual Vol. I.