Lecture 2: Data Transfers: Architectures, Models and Optimizations

CR11 – Hardware Compilation and Simulation
ENS-Lyon – M2IF

Christophe Alias
Outline

1. Architecture (recalls)
2. Roofline model
3. Loop transformations
Simple model

Transfer time: $T(n) = \lambda + n/BW$
Simple model

Transfer time: \( T(n) = \lambda + \frac{n}{BW} \)

Solutions

- Latency \( \lambda \): hardware prefetch
Simple model

Transfer time: $T(n) = \lambda + n/BW$

Solutions

- Latency $\lambda$: hardware prefetch
- Bandwidth $BW$: caches

→ Data reuse
The RAM is divided in lines of bits
The lines are grouped in banks
A data access consumes energy and latency

→ The data are sent by line (burst)
Architecture of a cache (direct-mapped)

- The data are read/write by burst
  → spatial reuse
- The tag allows to disambiguize the address ⊗
register byte s;
byte*a;
for i := 1 to 512/8
  s = s + a[i];

register byte s;
byte*a;
for i := 1 to N
  s = s + a[i] + a[i+4*512/8];

&a is aligned on 512 bits.

How many cache misses?
We maintain several banks

→ associativity = number of banks

Choose the first bank with a free line

If no bank is available, drop a line.
Operational Intensity

Rate of cache reuse:

\[ OI = \frac{\# \text{computations}}{\# \text{communications}} \]

- Computation: arithmetic operation
- Communication: cache miss
  - → depends on the cache architecture
- The cache model is often simplified (no associativity, line of size 1).
  - → quantifies the temporal reuse
Quizz

for $i := 1$ to $10$
for $j := 0$ to $N - 1$
    $a[j] = a[j] + 1$;

for $j := 0$ to $N - 1$
for $i := 1$ to $10$
    $a[j] = a[j] + 1$;

Cache w/o associativity, line of size 1, cache of size $\ell$

Give the operational intensity when $2\ell < N$

Give the operational intensity when $\ell \geq N$
for $i := 1$ to $10$
  for $j := 0$ to $N - 1$
    $a[j] = a[j] + 1;$

for $j := 0$ to $N - 1$
  for $i := 1$ to $10$
    $a[j] = a[j] + 1;$

Cache w/o associativity, line of size 1, cache of size $\ell$

Give the operational intensity when $2\ell < N$
→ Depends on the computation organization

Give the operational intensity when $\ell \geq N$
→ ratio volume(computation) / volume(data)
Roofline model

\[ \text{Perf}(OI) \leq OI \times BW \]
Roofline model

\[ \text{Perf}(OI) \leq OI \times BW \]

→ communication bounded
Roofline model

\[ \text{Perf}(OI) \leq OI \times BW \]

\[ \text{Perf}(OI) \leq PP \text{ (Peak Performance)} \]

→ communication bounded

→ compute bounded
Roofline model

More parallelism (PP →) → optimization required

HBM2 → DDR4 (BW →) → under used!

\( \text{IO}^* \)
More parallelism ($PP \nearrow$)

→ optimization required
Roofline model

- More parallelism (PP ↗)
- HBM2 → DDR4 (BW ↘)

→ optimization required

→ under used!
Optimizing operational intensity

for $i := 1$ to $10$
  for $j := 0$ to $N - 1$
    $a[j] = a[j] + 1$;
  permutation

for $j := 0$ to $N - 1$
  for $i := 1$ to $10$
    $a[j] = a[j] + 1$;

Loop permutation

→ reduces the reuse distance
Optimizing operational intensity

for $i := 1$ to $10$
  for $j := 0$ to $N - 1$
    $a[j] = a[j] + 1$;

for $i := 1$ to $N$
  for $j := 1$ to $N$
    $c[j] := a[i] + b[j]$;
  for $j := 1$ to $N$
    $d[i] := d[i] + c[j]$;

for $j := 0$ to $N - 1$
  for $i := 1$ to $10$
    $a[j] = a[j] + 1$;

for $i := 1$ to $N$
  for $j := 1$ to $N$
    $c[j] := a[i] + b[j]$;
  for $j := 1$ to $N$
    $d[i] := d[i] + c[j]$;

- Loop permutation
- Loop fusion

$\rightarrow$ reduces the reuse distance
Use case: matrix transposition

Hypothesis: cache 1-way, $\ell$ lines of $b$ bytes, write through, $N, P \gg b$

<table>
<thead>
<tr>
<th>optimization</th>
<th>reads</th>
<th>writes</th>
<th>$OI$</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>$NP$</td>
<td>$(N/b)P$</td>
<td>$b/(b+1)$</td>
</tr>
<tr>
<td>permutation</td>
<td>$N(P/b)$</td>
<td>$NP$</td>
<td>$b/(b+1)$</td>
</tr>
</tbody>
</table>
Use case: matrix transposition

Hypothesis: cache 1-way, $\ell$ lines of $b$ bytes, write through, $N, P \gg b$

<table>
<thead>
<tr>
<th>optimization</th>
<th>reads</th>
<th>writes</th>
<th>$OI$</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>$NP$</td>
<td>$(N/b)P$</td>
<td>$b/(b+1)$</td>
</tr>
<tr>
<td>permutation</td>
<td>$N(P/b)$</td>
<td>$NP$</td>
<td>$b/(b+1)$</td>
</tr>
</tbody>
</table>
**Use case: matrix transposition**

Hypothesis: cache 1-way, $\ell$ lines of $b$ bytes, write through, $N, P \gg b$

<table>
<thead>
<tr>
<th>optimization</th>
<th>reads</th>
<th>writes</th>
<th>$OI$</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>$NP$</td>
<td>$(N/b)P$</td>
<td>$b/(b+1)$</td>
</tr>
<tr>
<td>permutation</td>
<td>$N(P/b)$</td>
<td>$NP$</td>
<td>$b/(b+1)$</td>
</tr>
</tbody>
</table>

byte $A[N][P], B[P][N]$;
for $i := 0$ to $P - 1$
for $j := 0$ to $N - 1$
**Use case: matrix transposition**

```
byte A[N][P], B[P][N];
for i := 0 to P - 1
  for j := 0 to N - 1
```

Hypothesis: cache 1-way, $\ell$ lines of $b$ bytes, write through, $N, P \gg b$

<table>
<thead>
<tr>
<th>optimization</th>
<th>reads</th>
<th>writes</th>
<th>$OI$</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>$NP$</td>
<td>$(N/b)P$</td>
<td>$b/(b+1)$</td>
</tr>
<tr>
<td>permutation</td>
<td>$N(P/b)$</td>
<td>$NP$</td>
<td>$b/(b+1)$</td>
</tr>
</tbody>
</table>
Use case: matrix transposition

Hypothesis: cache 1-way, $\ell$ lines of $b$ bytes, write through, $N, P \gg b$

<table>
<thead>
<tr>
<th>optimization</th>
<th>reads</th>
<th>writes</th>
<th>$OI$</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>$NP$</td>
<td>$(N/b)P$</td>
<td>$b/(b+1)$</td>
</tr>
<tr>
<td>permutation</td>
<td>$N(P/b)$</td>
<td>$NP$</td>
<td>$b/(b+1)$</td>
</tr>
<tr>
<td>tiling</td>
<td>$N(P/b)$</td>
<td>$(N/b)P$</td>
<td>$b/2$</td>
</tr>
</tbody>
</table>

for $i := 0$ to $P - 1$
for $j := 0$ to $N - 1$
Use case: matrix transposition

Hypothesis: cache 1-way, $\ell$ lines of $b$ bytes, write through, $N, P \gg b$

<table>
<thead>
<tr>
<th>optimization</th>
<th>reads</th>
<th>writes</th>
<th>$OI$</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>$NP$</td>
<td>$(N/b)P$</td>
<td>$b/(b+1)$</td>
</tr>
<tr>
<td>permutation</td>
<td>$N(P/b)$</td>
<td>$NP$</td>
<td>$b/(b+1)$</td>
</tr>
<tr>
<td>tiling</td>
<td>$N(P/b)$</td>
<td>$(N/b)P$</td>
<td>$b/2$</td>
</tr>
</tbody>
</table>
Loop tiling

for $i := 0$ to $N - 1$
  for $j := 0$ to $P - 1$

for $i := 0$ to $N - 1$ step $\ell$
  for $j := 0$ to $P - 1$ step $b$
    for $ii := i$ to $\min\{i + b - 1, N - 1\}$
      for $jj := j$ to $\min\{j + \ell - 1, P - 1\}$

- Partition the computation in atomic tiles
- Tiles tend to access a different pool of data → locality & parallelization
- “The” transformation