Linear Scheduling

\[ \text{for } i := 1 \text{ to } N \]
\[ \text{for } j := 1 \text{ to } N \]
\[ B: \quad a[j] := a[j] + 1; \]

▶ Scope: perfect loop nests with uniform dependences
▶ \( \theta : D_B \rightarrow (\mathbb{Z}^d, \ll) \quad \vec{i} \mapsto A\vec{i} \)

Latency

\[ \text{for } i := 1 \text{ to } N \]
\[ \text{for } j := 1 \text{ to } N \]
\[ B: \quad a[j] := a[j] + 1; \]

▶ Latency: “Number of steps”
▶ \( \lambda = \text{card}\{\theta(\vec{i}), \vec{i} \in D_B\} \)
Quizz

for $i := 1$ to $N$
for $j := 1$ to $N$
B: \[ a[j] := a[j] + 1; \]

Give the latency for:
- $\theta_1(i,j) = i$
- $\theta_2(i,j) = i + j$
- $\theta_3(i,j) = (i + j, i)$

Theorem 1
If $d = 1$, then the latency is an affine form of program parameters

Monodimensional case ($d = 1$)

for $i := 1$ to $N$
for $j := 1$ to $N$
B: \[ a[i,j] := a[i-1,j] + a[i,j-1]; \]

- $\theta(\vec{\iota}) = A\vec{\iota} = \vec{\tau} \cdot \vec{\iota}$
- Correctness: $\vec{\tau} \cdot \vec{d} > 0 \quad \forall \vec{d} \in \Delta D$

Quizz

for $i := 1$ to $N$
for $j := 1$ to $N$
B: \[ a[j] := a[j] + 1; \]

Characterize all the correct linear schedules

Limits

for $i := 1$ to $N$
for $j := 1$ to $N$
B: \[ s := s + a[i,j]; \]

Prove that no monodimensional linear schedule exists
Multidimensional case ($d \geq 2$)

\[
\theta(i) = \left( \begin{array}c \vec{r}_1 \\
... \\
\vec{r}_d \end{array} \right)_i
\]

Quizz

Give $\vec{r}_1$ and $\vec{r}_2$ for $\theta(i,j) = (i,j)$

Finding $\vec{r}_1$

\[
\forall \vec{d} \in \Delta D \quad \vec{r} \cdot \vec{d} > 0 \text{ or } \vec{r} \cdot \vec{d} = 0
\]
Finding $\bar{\tau}_2$

\[ \text{for } i := 1 \text{ to } N \]
\[ \text{for } j := 1 \text{ to } N \]
\[ B: \quad s := s + a[i,j]; \]
\[ \Delta D' = \{(0, 1)\} \]

- Focus on respected dependences $\Delta D'$
- Pick $\bar{\tau}_2 \in \text{cone}(\Delta D')^*$

Algorithm

\[ \text{for } i := 1 \text{ to } N \]
\[ \text{for } j := 1 \text{ to } N \]
\[ B: \quad s := s + a[i,j]; \]
\[ i := 1 \]
\[ \text{while } \Delta D \neq \emptyset \]
\[ C_T := \text{cone}(\Delta D)^* \]
\[ \text{Pick } \bar{\tau}_i \in C_T \setminus \{0\} \]
\[ \text{Remove } \text{from } \Delta D \text{ the satisfied dependences } (\bar{\tau}_i \cdot \bar{d} > 0) \]
\[ i := i + 1 \]

Correctness

Lemma 2
\[ \theta \text{ is correct over } \Delta D \text{ iff } \theta(\bar{d}) \gg \bar{0} \quad \forall \bar{d} \in \Delta D \]

Quizz: Prove it

Theorem 3
The algorithm finds a correct schedule

Proof.
Consider $\bar{d} \in \Delta D$. If the algorithm terminates,
\[ \exists k : (\bar{\tau}_k \text{ satisfies } \bar{d}) \land (\bar{\tau}_k \text{ respects } \bar{d} \quad \forall \ell < k) \]
Hence, $\theta(\bar{d}) \gg \bar{0}$.

How to “pick” $\bar{\tau}_i$?

\[ \text{for } i := 1 \text{ to } N \]
\[ \text{for } j := 1 \text{ to } N \]
\[ B: \quad a[j] := a[j] + 1; \]

Greedy: satisfy as much dependences as possible
How to “pick” $\vec{r}_i$?

```
forpar j := 1 to N
   for i := 1 to N
      B: a[j] := a[j] + 1;
```

Lazy: satisfy as less dependences as possible

Lessons

- **Greedy approach:** minimize the dimension (sequentiality degree) → maximize the parallelism degree
- **Lazy approach:** push dependence resolution into inner loops → improve the data locality
- **The lazy approach is the best** → parallel outer loops + inner loops with good locality

Lab

Consider the Seidel 1D kernel:

```
for t := 1 to T
   for i := 1 to N
      a_i = a_{i-1} + a_i + a_{i+1}
```

1. Give $\Delta D$, the dependence cone and the time cone.
2. Apply the linear scheduling algorithm with the greedy approach.
3. Draw the iteration domain (for small values of $T$ and $N$) and the obtained execution order.