Modelling execution time – an example

Atmosphere model

- simulates atmospheric processes
- solves a set of partial differential equations
- The behavior of these equations on a continuous space is approximated by their behavior on a finite set of regularly spaced points in that space
Partial differential equations in the atmosphere model

Conservation of momentum:
\[
\frac{du}{dt} - \left( f + u \frac{tan\phi}{a} \right) u = - \frac{1}{\rho \cos\phi} \frac{\partial p}{\partial \lambda} + F_\lambda
\]
\[
\frac{du}{dt} + \left( f + u \frac{tan\phi}{a} \right) u = - \frac{1}{\rho \phi} \frac{\partial p}{\partial \phi} + F_\phi
\]

Hydrostatic approximation:
\[
g = - \frac{1}{\rho} \frac{\partial p}{\partial z}
\]

Conservation of mass:
\[
\frac{\partial p}{\partial t} = - \frac{1}{a \cos\phi} \left( \frac{\theta}{\theta \lambda} (\rho u) + \frac{\theta}{\theta \phi} (\rho u \cos\phi) \right) - \frac{\theta}{\theta z} (\rho u)
\]

Conservation of energy:
\[
C_p \frac{dT}{dt} - \frac{1}{\rho} \frac{dp}{dt} = Q
\]

State equation (atmosphere):
\[
p = \rho RT
\]
Communication pattern

Each point uses the nine-point stencil to calculate its horizontal motion and uses the three-point stencil to calculate its vertical motion.

Fig 1. Stencils used in the atmosphere model

Fig 2. Communication pattern for the computation with nine-point stencil
If we assume a grid of size N*N*Z points, and using 1-D decomposition (along y-axis) to partition the grid among P processors, then

- each task is responsible for a subgrid of size N*(N/P)*Z
- then, \( T_{\text{comp}} \) for each subgrid can be calculated as follows, where \( t_c \) is the average time of calculating a single grid point

\[
T_{\text{comp}} = t_c \times N \times (N/P) \times z
\]
Modelling the time of sending messages

- $T_{msg}$, denoted the time used to send a message, can be calculated as follows, where $t_s$ is the message startup time, $t_w$ is the transfer time per word (or byte), $L$ is the size of the message:

$$T_{msg} = t_s + t_w L$$

![Diagram showing the relationship between time, startup cost, transfer cost per word, and message length.](image)
Modelling communication time

- Communication time for calculating a subgrid can be computed as follows

\[ T_{\text{comm}} = 2(t_s + tw)2NZ \]

- Hence, the performance model for the execution time of calculating the velocity of the grid of points is

\[ T_p = T_{\text{comp}} + T_{\text{comm}} = t_c N^* (N/p) Z + (t_s + tw)2NZ \]

- From the performance model of the execution time, we can know that:
  - Execution time decreases with increasing P but is bounded by the cost of transferring messages
  - Execution time increases with increasing N, Z, \( t_c \), \( t_s \), \( tw \)
**Speedup and parallel efficiency**

- The execution time on one processor is
  \[ T_1 = t_c N^2 Z \]

- Speedup is
  \[ S(P) = \frac{t_c N^2 Z P}{t_c N^2 Z + 2 t_s P + t_w 4 N Z P} \]

- Parallel efficiency can be calculated as
  \[ E = \frac{t_c N^2 Z}{t_c N^2 Z + 2 t_s P + t_w 4 N Z P} \]
 Iso-efficiency

\[ E = \frac{t_c N^2 Z}{t_c N^2 Z + 2t_s P + t_w 4NZP} \]

can be reduced to

\[ E = \frac{t_c}{t_c + \frac{t_s 2P}{ZN^2} + \frac{t_w 4P}{N}} \]

When N=P, E remains approximately constant as P changes (except when P is small)
If applying 2-D decomposition, then

- Each task is responsible for \((N/\sqrt{P}) \times (N/\sqrt{P}) \times 2\) points
- Execution time can be modelled as

\[
T_{\text{comp}} = t_c \frac{N^2 Z}{P}
\]
\[
T_{\text{comm}} = 4(t_s + t_w) \frac{2N}{\sqrt{P}} Z
\]
\[
T_P = T_{\text{comp}} + T_{\text{comm}} = t_c \frac{N^2 Z + t_s 4P + t_w 8NZ \sqrt{P}}{P}
\]
Iso-efficiency

- Parallel efficiency can be modelled as

\[ \frac{t_c N^2 Z}{t_c N^2 Z + t_s 4P + t_w 8NZ \sqrt{P}} \]

- When \( N = \sqrt{P} \), \( E \) remains constant as \( P \) increases

- Therefore, the algorithm will have better scalability when applying 2D decomposition than 1D decomposition

Iso-efficiency in 1D

\( N = P \)
Modelling in the real systems

- We made some assumptions when we construct the above models

- In real systems:
  - Startup time may increase for very big messages
  - Transfer time per unit of message may vary because of bandwidth competition
  - Communication may overlap

\[ T_{\text{comm}} = 2(t_s + t_w2N) \]
Decomposition analysis

- boundary surfaces between sub-grids are shaded.
- Boundary surfaces need typically to be communicated.
- The lower surface-to-volume ratio, the better:
  - Surface = communication
  - Volume = computation
Decomposition analysis

Consider a 3-D grid and assume the grid is a cube

- Volume $V = c \times n$, where $c$ = number of cells per PE, $n$ is the number of processors
- The length of the grid in each dimension is $V^{1/3}$
## Decomposition analysis

<table>
<thead>
<tr>
<th>Sub-grid Volume</th>
<th>Sub-grid Length</th>
<th>Sub-grid Surfaces</th>
<th>Surface to Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D</td>
<td>c</td>
<td>( V^{1/3} / n )</td>
<td>( 2c^{2/3} \cdot n^{2/3} )</td>
</tr>
<tr>
<td>2-D</td>
<td>c</td>
<td>( V^{1/3} / n^{1/2} )</td>
<td>( 4c^{2/3} \cdot n^{1/6} )</td>
</tr>
<tr>
<td>3-D</td>
<td>c</td>
<td>( V^{1/3} / n^{1/3} )</td>
<td>( 6c^{2/3} )</td>
</tr>
</tbody>
</table>

### Diagrams:
- **1-D**
- **2-D**
- **3-D**