Case Study: Specialized Serial vs. Generalized Parallel Processing

- Applications arrive to a loan application office according to a Poisson process with rate $\lambda=1/1.25$ applications/hour

- The processing of each application required Check credit, prepare covenant, price loan, disburse funds

- The office has four employees: Alfie, Betty, Chuck, and Doris.
Case Study: Specialized Serial vs. Generalized Parallel Processing

- Assume that employees are allocated “in series” with each employee handling one step of the loan processing.

- Every employee handle one step of the loan processing steps in exponentially distributed service time with mean=1 hour.
Arrivals according to a Poisson process with rate $\lambda = \frac{1}{1.25}$ applications/hour, exponentially distributed service time with mean $= 1$ hour

- Run for 160 hours
Case Study: Model 3-2, Specialized Serial Processing (cont’d.)

- Four Process modules
- Four separate Resources
  - Expo process time: Expression (via Expression Builder)
- Default entity picture (report) is OK
- Default Resource animations almost OK
  - Make Idle picture same as Busy
  - Select correct Resource name in Identifier field
- Queue, Resource data modules OK
- Plot WIP – use Expression builder to find EntitiesWIP(Application)
  - Fixed Y axis max = 25 to compare with next three models
- Fill in Run > Setup, lengthen queue animations
- Watch avg, max no. applications in process (WIP); avg, max total time in system of applications.

- Viewing Report

<table>
<thead>
<tr>
<th>serial</th>
<th>Total WIP</th>
<th>Total Time in System</th>
<th>Total Waiting Time</th>
<th>Number Processed</th>
<th>Avg. Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg. 12.39 Max. 21</td>
<td>Avg. 16.08 Max. 27.21</td>
<td>Avg. 11.98 Max. 22.27</td>
<td>117</td>
<td>0.78</td>
</tr>
</tbody>
</table>
What’s wrong with series model?

1) Long waiting time.
2) Wasted capacity (Betty’s idle but Chuck and Doris are overloaded).

How can we improve it?

Use parallel system where each employee can handle any of the four loan processing steps.
Case Study: Model 3-3, Generalized Parallel Processing

• Create, Dispose, plot, Run > Setup almost same
  ▪ Just change some labels, etc.
• Each employee can handle any of the four loan processing steps in exponentially distributed service time with mean=1 hour
Case Study: Model 3-3, Generalized Parallel Processing (cont’d.)

- Replace four earlier Process modules with just a single Process module
  - One Resource (Loan Officer), but four units of it
  - Still set Quantity to 1 since application just needs 1 officer
  - Delay type – Expression
    \[ \text{EXPO}(1) + \text{EXPO}(1) + \text{EXPO}(1) + \text{EXPO}(1) \]
    - Why not \(4 \cdot \text{EXPO}(1)\)?

- Modify Resource Animation for four units
  - Open Model 3-2 Resource Animation to get Resource Picture Placement window, open Idle picture
  - Duplicate white square three times, realign; copy to Busy
  - In model window, double-click Seize Area, then Add three
  - Still not completely accurate animation (order) – need Sets
Case Study: Compare Model serial vs. parallel

<table>
<thead>
<tr>
<th>Model</th>
<th>Total WIP</th>
<th>Total Time in System</th>
<th>Total Waiting Time</th>
<th>Number Processed</th>
<th>Avg. Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-2 (serial)</td>
<td>12.39</td>
<td>21</td>
<td>16.08</td>
<td>27.21</td>
<td>117</td>
</tr>
<tr>
<td>3-3 (parallel)</td>
<td>4.61</td>
<td>10</td>
<td>5.38</td>
<td>13.73</td>
<td>135</td>
</tr>
</tbody>
</table>

In parallel model, the waiting time decreases from 11.98 to 1.33 hours.