1. What is the speedup achieved when 85% of Computation can be parallelized using 7 processors?

Given:

85% of Computation can be parallelized
15% of Computation is serial execution.

$$S(p,n) = \frac{1}{f + \frac{(1-f)}{p}} = \frac{1}{0.15 + \frac{(1-0.15)}{7}} = \frac{1}{0.15+0.121}$$

$$S(p,n) = \frac{1}{0.271} = 3.69$$

2. Given: Execution time is 120 seconds
problem size = 1
Initiation phase = 20 seconds

Let $T_{1,n}$ be sequential execution time

$C_s = 20$, C_s be inherently sequential execution cost
$C_p = 100$, C_p be parallel execution cost

$$T_{1,n} = C_s + \left[(C_p \cdot n^2) \right] \boxed{\text{ie, } C_s + C_p n^2} \quad \text{——— (1)}$$

$$T_{p,n} = C_s + \left[\frac{(C_p \cdot n^2)}{p} \right] \boxed{\text{ie., } C_s + \frac{C_p n^2}{p}} \quad \text{——— (2)}$$

a) $$S(p,n) = \frac{T_{1,n}}{T_{p,n}} = \frac{C_s + C_p n^2}{C_s + \frac{C_p n^2}{p}} \quad \text{——— (3)}$$

$$\Rightarrow S(p,n) = \frac{20 + 100n^2}{20 + \frac{100n^2}{p}}$$

b) for problem size 1 ($n=1$) and processors ($p=5$) equation (3) becomes

$$S(5,1) = \frac{T_{1,1}}{T_{5,1}} = \frac{20+100(1)}{20+\frac{100(1)}{5}} = \frac{120}{40} = 3$$

Speedup $\boxed{S(5,1)=3}$

c. Execution time of the application if problem size is increased to 5 and it is run on 10 processors and on 20 processors

$n=5$

$$T_{10,5} = 20 + \frac{100(5\times5)}{10} = 20+250 = 270 \text{ seconds}$$

$n=5$

$$T_{20,5} = 20 + \frac{100(25)}{20} = 20+125 = 145 \text{ seconds}$$

d. Speedup for each measurements

$$S(10,5) = \frac{20+100(25)}{20+\frac{100(25)}{10}} = \frac{20+100(25)}{20+250} = \frac{2520}{270}$$

$$\boxed{S_{10,5} = 9.33}$$

$$S(20,5) = \frac{20+100(25)}{20+\frac{100(25)}{20}} = \frac{20+100(25)}{20+125} = \frac{2520}{145} = 17.38$$

$$\boxed{S_{20,5} = 17.38}$$

3) Consider an application where execution of floating point instructions on a certain processor P consumes 60% of total run time. Assuming 20% of the floating-point time is spent in square root calculations

a. All floating point instruction improved by factor of 1.5

Using Amdhal's law: $S_p(n) = \frac{1}{f + \frac{(1-f)}{p}}$

[inherently sequential 40%] $f = 0.4$, $p = 1.5$ $S_p(n) = \frac{1}{0.4 + \frac{0.6}{1.5}} = 1.25$

Alternatively speed up of square root operation is improved by factor of 8.

$f = 0.4 + 0.45$

[inherently sequential 85%] $f = .85$, $p = 8$

$$S_p(n) = \frac{1}{0.85 + \frac{(1-0.85)}{8}} = \frac{1}{0.86875} = 1.15$$

Thus, the application would benefit the most when all floating point instruction improved by factor of 1.5

b. Speedup achieved on 20-CPU system, if 85% of code can be perfectly parallelized.

Using Amdhal's law

$$S_{20,n} = \frac{1}{0.15 + \frac{(1-0.15)}{20}} = \frac{1}{0.15 + \frac{0.85}{20}} = \frac{1}{0.1925}$$

$$\boxed{S_{20,n} = 5.195}$$

c) The fraction of code that can be parallelized to achieve speed up of 10

Using Amdhal's law

$$\frac{1}{f + \frac{(1-f)}{20}} = 10$$

$$\frac{1}{\frac{20f + (1-f)}{20}} = 10 \Rightarrow 10\left[\frac{20f + (1-f)}{20}\right] = 1$$

$$200f + 10 - 10f = 20$$

$$190f = 10$$

$$f = \frac{10}{190} = 0.05$$

ie., 5% of Code is serial execution

$$\boxed{\text{Hence, 95\% of code is parallelizable.}}$$