side-effect

- no side-effect

Math Reduce

- $f(x) = \frac{9}{y}$

- Why?

- List

- Procedure

- $\text{map} (x)$

- $\text{reduce}$

- $\text{filter}$

- find

- map

- reduce

- $\text{reduce}$

- (map) for data reduction

- map reduce sequence - model

- functions

- make

- complex programs

- (many) for data manipulation

- CSE 535 Fall 05
1. $P_i = \text{sum}_i^2$

2. $\text{sum} = \text{sum} + 2$  \[ \text{sum} = 0 + 1 \]

3. $\text{sum} = \text{sum} + 4\text{element}$

A: [1 2 3 4]

- $\text{sum} = \text{sum} + x_i$

- If $i > n$

- Depend on $x_i$

- Sum of $x_i$

- Parallel execution (comp.)

- Case 2 is better performance than Case 1

- Memory

- F"r"ed
Main: create Process on P, & process on P2

\[ \text{sum,} = 0 \]
\[ \text{for } i = 1 \text{ to } n/2 \]
\[ \text{sum,} = \text{sum,} + A[i] \] on P1

\[ \text{sum,} = 0 \]
\[ \text{for } i = n/2+1 \text{ to } n \]
\[ \text{sum,} = \text{sum,} + A[i] \] on P2

\[ \rightarrow \text{sum} = \text{sum}, + \text{sum,} \]

\[ \text{wait for P, & P2 to finish.} \]

\[ \text{barrier syn.} \]

```
paral.isation is not easy
- all the dependences (data, control flow dependence)
  have to honoured or preserved
- added overhead in creating threads
  + data partitioning
  + synchronisation
- Has the performance improved?
```
\[
S(n) = \frac{T_2(n)}{T_1(n)}
\]

\[S > 5 \quad \Rightarrow \quad T_1(n) > T_2(n)\]

Ex. Problem:

\[
\text{Speed} = \frac{S}{\text{Time for # ex.}} = \frac{1}{T_1(n)}
\]

\[T_1 > T_2\]

\[\text{ex. time on the process}\]

\[T_1: \text{execute time on one processor}\]
Good news  

\[
\frac{d}{dt} \frac{f_5 + (1 - f_5)}{1} = \frac{d}{dt} \frac{f_5 + (1 - f_5)}{f_5 + (1 - f_5)}
\]

\[p \rightarrow \infty \]

\[\lim_{t \to \infty} = \frac{f_5}{1} = \frac{5}{1} = 5 \]

Taylor Law.
time-dependent

...new things & bugs - face condition.

- good pull depends upon 1+ 1 programs.

...syn. memory management etc.

- stress more nowadays may be due for dis. 

- performance becoming more complex & demanding.

...research on performance is not easy.

Developing periods programs which are...
HPR

Synchronization, fault tolerance, performance in symmetric parallel systems, management, scalability

Execution model / Runtime System/Less
Explain, personal, your performace. Does this with my resamith.

Also, give me the system this.'