

```

69         var wh = w.height();
70         var tw = t.width();
71         var ww = w.width();
72
73         if (y + th + ay >= b &&
74             y <= b + wh + ay &&
75             x + tw + ax >= a &&
76             x <= a + ww + ax) {
77
78             //trigger the custom event
79             if (!t.appeared) t.trigger('appear', settings.data);
80
81         } else {
82
83             //it scrolled out of view
84             t.appeared = false;
85
86         }
87     };

```



## Parallelism & Concurrence Part 2

### Descomposición de problemas y Especificación Formal

#### 1. Descomposición de problemas:

- **Descomposición Funcional:** Divide el cómputo en sub-cálculos independientes o en fases con dependencias jerárquicas.
- **Descomposición Geométrica:** Divide el cálculo en secciones físicas o lógicas del sistema, balanceando la carga según una regla regular o aleatoria.
- **Descomposición Iterativa:** Aplica repetidamente operaciones sobre datos de manera simultánea.

#### 2. Metodología Básica (PCAM):

- **Partición:** Descomponer el cómputo y los datos en pequeñas tareas.
- **Comunicación:** Determinar las estructuras de comunicación necesarias entre tareas.
- **Aglomeración:** Analizar y combinar tareas para optimizar prestaciones y reducir costos de comunicación.
- **Mapping:** Asignar tareas a procesadores/nodos para maximizar la utilización y minimizar costos de comunicación.

#### 3. Mapeo de Procesos:

- Ubicar las tareas concurrentes en distintos procesadores mejora la concurrencia.
- Ubicar las tareas que se comunican con frecuencia en el mismo procesador mejora la localidad de los datos.

El mapping es importante para el rendimiento de los algoritmos paralelos. Se debe encontrar un equilibrio óptimo para el rendimiento paralelo.

#### 4. Efecto de la Granularidad en el Rendimiento:

- **Scaling Down:** Disminuir el número de procesadores puede mejorar el rendimiento asignado más de un procesador virtual.

Siendo:

$$P = \text{Procesadores Reales}$$

$$N = \text{Procesadores Virtuales}$$

$N/P \Rightarrow$  es el factor de aumento de los cálculos a realizar en cada procesador físico.

## Especificación Formal de Sistemas Concurrentes

#### 1. Identificación de Concurrencia

**Condiciones de Bernstein:**

- $L(Si) \cap E(Sj) = \emptyset$
- $E(Si) \cap L(Sj) = \emptyset$
- $E(Si) \cap E(Sj) = \emptyset$

#### 2. Notaciones para Especificar Concurrencia

- a. **Grafos de Precedencia:** Notación gráfica para representar instrucciones concurrentes.

- **Nodo:** Representa un conjunto de instrucciones.
- **Flecha:** Indica precedencia entre nodos.

- b. **Sentencias COBEGIN-COEND:** Bloques de instrucciones que pueden ejecutarse concurrentemente.

#### 3. Ejemplo de Aplicación de Condiciones de Bernstein

- a. Supongamos los conjuntos de instrucciones:

- $S1 \rightarrow a := x + y$
- $S2 \rightarrow b := z - 1$
- $S3 \rightarrow c := a - b$
- $S4 \rightarrow w := c + 1$

- b. Debe observarse cuáles pares de sentencias pueden ejecutarse concurrentemente siguiendo las condiciones de Bernstein.

## Conceptos Claves

- **Descomposición Funcional, Geométrica e Iterativa:** Diferentes estrategias para dividir problemas en tareas paralelas.
- **PCAM:** Metodología de partición, comunicación, aglomeración y mapeo.
- **Condiciones de Bernstein:** Criterios para determinar la concurrencia entre conjuntos de instrucciones.
- **Grafos de Precedencia:** Herramienta visual para planificar la ejecución concurrente.
- **Sentencias COBEGIN-COEND:** Estructura para definir bloques de instrucciones concurrentes.

## Resumen Gráficos

### Sentencias COBEGIN-COEND:

Este bloque indica que las instrucciones dentro de `cobegin` y `coend` pueden ejecutarse en cualquier orden, en paralelo.

```
(cobegin  
  (S1)  
  (S2)  
coend)
```

### Grafos de Precedencia:

En este grafo, cada nodo representa un conjunto de instrucciones, las flechas indican que una instrucción debe completarse antes de que la siguiente comience y si aparecen dos conjuntos de instrucciones en paralelo significa que pueden ejecutarse concurrentemente.

