

Técnicas de autooptimización en recorridos de árboles por medio de *backtracking*

Manuel Quesada Martínez

25 de febrero de 2009

Índice

- 1 Introducción
- 2 Técnica de backtracking
- 3 Recorridos secuenciales
 - Modelado de los recorridos secuenciales
 - Metodología para la estimación de los recorridos
 - Resultados experimentales (Selección de versiones)
- 4 Recorridos paralelos
 - Introducción a los recorridos paralelos
 - Esquemas algorítmicos paralelos
 - Parámetros, metodología y modelos
 - Resultados experimentales (Autoconfiguración de los parámetros)
- 5 Conclusiones y trabajos futuros

Introducción

- **Motivación:** abstraer a los usuarios de los problemas que plantea el paralelismo.
- **Toma de decisión:** objetivo es minimizar el tiempo de ejecución.
 - Seleccionando entre diferentes esquemas
 - Proponiendo valores para parámetros del algoritmo
- **Esqueletos algorítmicos:** conjunto de procedimientos que constituyen el armazón para resolver un problema dado, programando una técnica particular. Facilitan la programación de los usuarios no expertos.

Introducción

- **Motivación:** abstraer a los usuarios de los problemas que plantea el paralelismo.
- **Toma de decisión:** objetivo es minimizar el tiempo de ejecución.
 - Seleccionando entre diferentes esquemas
 - Proponiendo valores para parámetros del algoritmo
- **Esqueletos algorítmicos:** conjunto de procedimientos que constituyen el armazón para resolver un problema dado, programando una técnica particular. Facilitan la programación de los usuarios no expertos.

Introducción

- **Motivación:** abstraer a los usuarios de los problemas que plantea el paralelismo.
- **Toma de decisión:** objetivo es minimizar el tiempo de ejecución.
 - Seleccionando entre diferentes esquemas
 - Proponiendo valores para parámetros del algoritmo
- **Esqueletos algorítmicos:** conjunto de procedimientos que constituyen el armazón para resolver un problema dado, programando una técnica particular. Facilitan la programación de los usuarios no expertos.

Introducción

- **Modelo teórico** del tiempo de ejecución
 - Aparecen parámetros en el modelo que debemos calcular su valor
 - Existen dos tipos de parámetros: parámetros del sistema (**SP**) y parámetros del algoritmo (**AP**)
 - Cálculo de los *SP* en tiempo de **instalación** y de los *AP* en tiempo de **ejecución**

$$T. \text{ Ejecución}(n) = f(n, SP, AP)$$

- **Problemas:** dependencias entre los parámetros y limitaciones del tiempo de decisión.

$$T. \text{ Ejecución}(n) = f(n, g(n, AP), AP)$$

Introducción

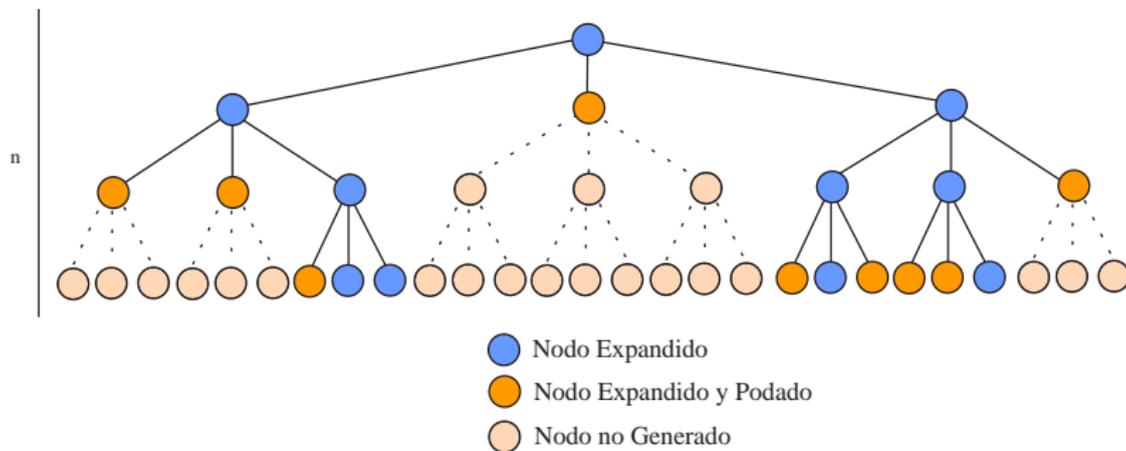
- **Modelo teórico** del tiempo de ejecución
 - Aparecen parámetros en el modelo que debemos calcular su valor
 - Existen dos tipos de parámetros: parámetros del sistema (**SP**) y parámetros del algoritmo (**AP**)
 - Cálculo de los *SP* en tiempo de **instalación** y de los *AP* en tiempo de **ejecución**

$$T. \text{ Ejecución}(n) = f(n, SP, AP)$$

- **Problemas:** dependencias entre los parámetros y limitaciones del tiempo de decisión.

$$T. \text{ Ejecución}(n) = f(n, g(n, AP), AP)$$

Técnica de *backtracking*



Modelado de los recorridos secuenciales

- **Modelo de tiempo teórico:**

$$TE(n) = k \cdot NNG \cdot TCN$$

- **Parámetros:**

- Número de nodos generados (NNG)
- Tiempo de cómputo de un nodo (TCN)
- Porcentaje de nodos generados (k)
 - Depende del **tamaño** de la entrada
 - Depende de la **naturaleza** de la entrada

- **Modelo de tiempo teórico considerando distintos niveles:**

$$TE = k \cdot \left(\sum_{i=1}^{NumNiveles} NNG_i \cdot TCN_i \right)$$

Modelado de los recorridos secuenciales

- **Modelo de tiempo teórico:**

$$TE(n) = k \cdot NNG \cdot TCN$$

- **Parámetros:**

- Número de nodos generados (NNG)
- Tiempo de cómputo de un nodo (TCN)
- Porcentaje de nodos generados (k)
 - Depende del **tamaño** de la entrada
 - Depende de la **naturaleza** de la entrada

- **Modelo de tiempo teórico considerando distintos niveles:**

$$TE = k \cdot \left(\sum_{i=1}^{NumNiveles} NNG_i \cdot TCN_i \right)$$

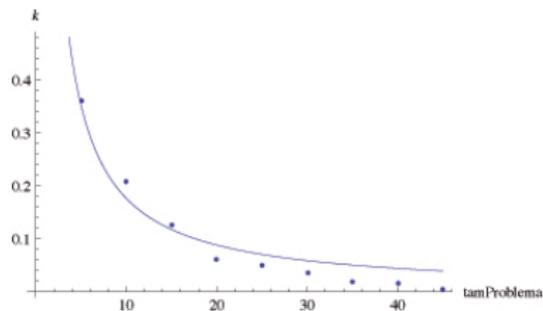
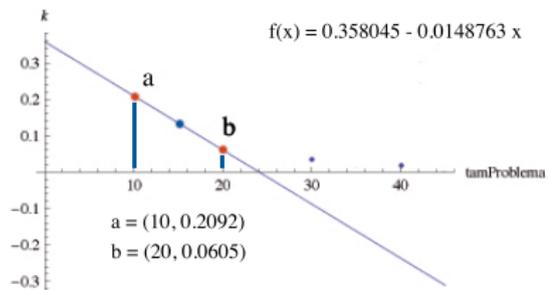
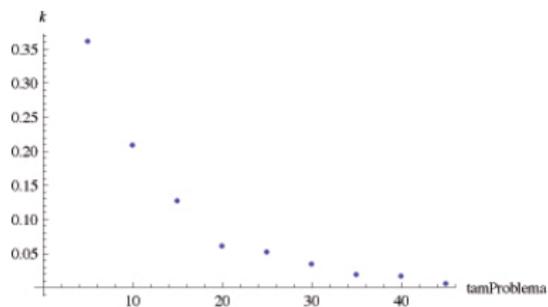
Modelado de los recorridos secuenciales

- Utilizaremos los modelos anteriores
- Técnicas para la estimación de los parámetros del modelo
 - **Tiempo de instalación:** podemos realizar pruebas genéricas que nos den información global del problema que vamos a resolver
 - **Tiempo de ejecución:** a partir de la entrada concreta podemos refinar la información de la instalación
- Diseño de una **metodología** general para la estimación de los recorridos secuenciales

Metodología para la estimación de los recorridos

- **Fase de instalación**
 - Estimación de TCN
 - Estimación general
 - Equivalente al tiempo de cómputo de las funciones del esquema
 - Estimación de k
 - Valor constante
 - Extrapolando valores por medio de una recta
 - Aproximando por medio de una función conocida

Metodología para la estimación de los recorridos



Metodología para la estimación de los recorridos

- **Fase de ejecución**

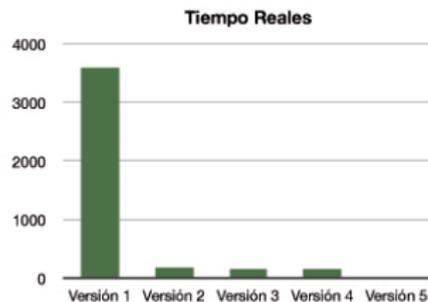
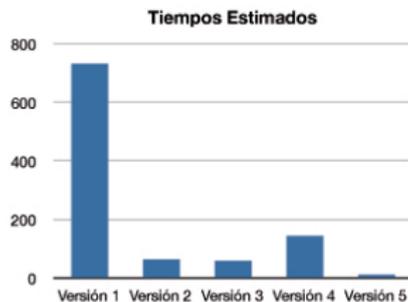
- Agrupando las entradas y generando problemas más pequeños
- Seleccionando valores aleatoriamente

$$k_{media} = 0.5 \cdot k_{TiempoInstalacion} + 0.5 \cdot k_{TiempoEjecucion}$$

Resultados experimentales (Selección de versiones)

● El problema de la Mochila 0/1

- Implementaremos **5 versiones** que resuelven el problema
- Evaluaremos las distintas formas de estimar los parámetros del modelo
 - Dificultad de ajustar los tiempos reales
 - Necesaria la información de la ejecución
 - El modelo **detecta tendencias** del tiempo de ejecución entre diferentes esquemas



Introducción a los recorridos paralelos

Recorridos paralelos de *backtracking*

- Para cada solución paralela aparecen **nuevos parámetros de configuración** para cada esquema. Importancia elegir valores apropiados para lograr beneficios en el tiempo de ejecución
- La programación de las soluciones paralelas dependen de las **plataformas** donde vayamos a ejecutar nuestros códigos (GPUs, Multicore, Cluster, Heterogéneos)
- Dificultad de abordar soluciones paralelas para programadores secuenciales

Introducción a los recorridos paralelos

Objetivos

- Diseñar una metodología general para proponer valores de los parámetros a los usuarios
- Estudiar diferentes esquemas algorítmicos paralelos de *backtracking* y modelar su comportamiento
- Simplificar la programación paralela con un esquema algorítmico secuencial que se paralelice de forma transparente a los usuarios

Introducción a los recorridos paralelos

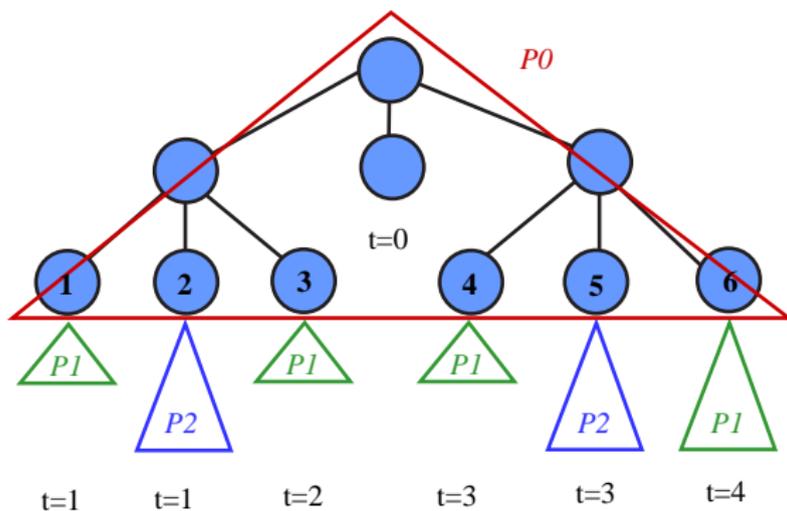
Objetivos

- Diseñar una metodología general para proponer valores de los parámetros a los usuarios
- Estudiar diferentes esquemas algorítmicos paralelos de *backtracking* y modelar su comportamiento
- Simplificar la programación paralela con un esquema algorítmico secuencial que se paralelice de forma transparente a los usuarios

Esquemas algorítmicos paralelos

- **EMEAE**: esquema maestro esclavo con asignación estática de tareas.
- **EMEAE+I**: esquema maestro esclavo con asignación estática de tareas y con intercambio de información.
- **EMEAD**: esquema maestro esclavo con asignación dinámica de tareas.

Esquemas algorítmicos paralelos



Esquema M/E asignación estática

```

1  ... ..
2
3  esquemaBacktracking( — , l, — , listaDeSubtrabajos );
4
5  int TS = length( listaDeSubtrabajos );
6
7  int * A = (int *) malloc (sizeof(int)*TS);
8  for ( int i = 0; i < TS; i++ ) A[i] = i mod p;
9
10 Cada Procesador idProc , para 0 < idProc < p {
11     for ( int j = 0 ; j < TS; j++ ) {
12         if ( A[j] == idProc ) {
13             struct nodoBacktraking * nodo;
14             nodo = getNodo(listaDeSubtrabajos , j);
15
16             esquemaBactracking( nodo , n-l , — , NULL);
17
18             free(nodo);
19         }
20     }
21 }
22
23  ... ..

```

Esquema M/E asignación dinámica

```

1  ... ..
2  esquemaBacktracking( — , l, — , bolsaDeSubtrabajos);
3
4  int TS = length(bolsaDeSubtrabajos);
5
6  Cada Procesador idProc, para 0 < idProc < p {
7      if ( idProc = 0 ) {
8          while ( TS > 0 ) {
9              recibeSolicitudDeTrabajo ( procDestino );
10             struct nodoBacktracking * nodo;
11             nodo = getNodo(bolsaDeSubtrabajos, TS);
12             enviaTrabajoARealizar ( procDestino, nodo, TS );
13             free(nodo); TS--;
14         }
15         for ( TS = 0; TS < p; TS++ ) {
16             enviaTrabajoARealizar ( TS, NULL, 0 );
17         }
18     }
19     else {
20         while ( trabajosPendientes != 0 ) {
21             enviaSolicitudDeTrabajo ( 0, idProc );
22             struct nodoBacktracking * nodo;
23             nodo = recibeTrabajoARealizar(procOrigen=0,trabajosPendientes);
24             if ( trabajosPendientes != 0 ) {
25
26                 esquemaBacktracking( nodo , n-l, — );
27
28             }
29             fee(nodo);
30         }
31     }
32 }
33 ... ..

```

Parámetros, metodología y modelos

- **Esquemas maestro esclavo:**
 - Profundidad del nivel inicial (**I**)
 - Número de procesadores a utilizar (**p**)
- **Esquemas con asignación estática:**
 - Tipo de asignación estática (**A**)
- **Esquemas con intercambio de información**
 - Porcentaje de intercambio de información (**e**)
- **Esquemas con asignación dinámica**
 - Número de subtrabajos a enviar (**t**)

Parámetros, metodología y modelos

- **Esquemas maestro esclavo:**
 - Profundidad del nivel inicial (**I**)
 - Número de procesadores a utilizar (**p**)
- **Esquemas con asignación estática:**
 - Tipo de asignación estática (**A**)
- **Esquemas con intercambio de información**
 - Porcentaje de intercambio de información (**e**)
- **Esquemas con asignación dinámica**
 - Número de subtrabajos a enviar (**t**)

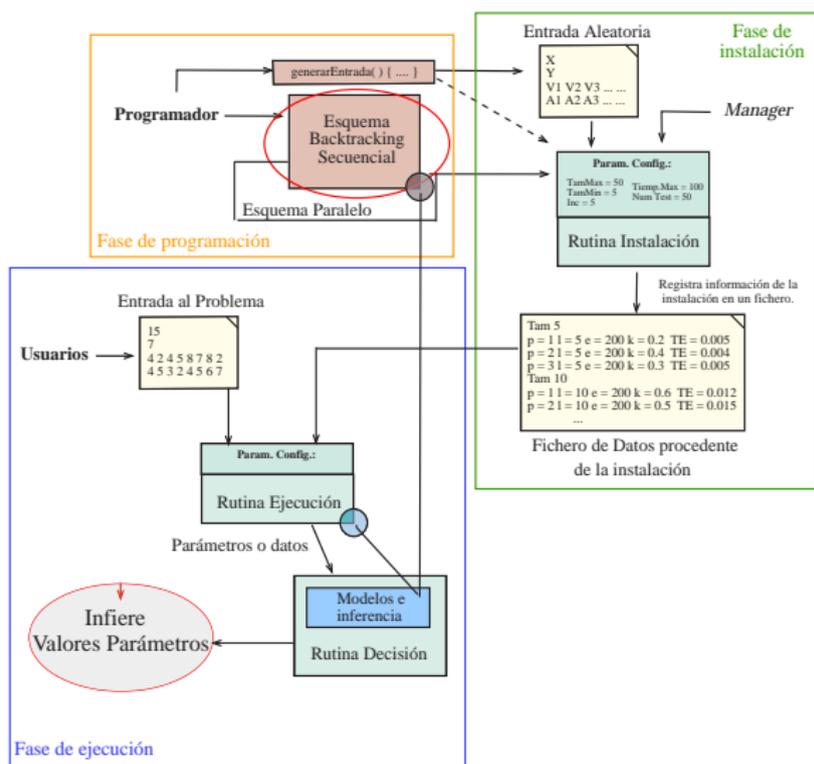
Parámetros, metodología y modelos

- **Esquemas maestro esclavo:**
 - Profundidad del nivel inicial (**I**)
 - Número de procesadores a utilizar (**p**)
- **Esquemas con asignación estática:**
 - Tipo de asignación estática (**A**)
- **Esquemas con intercambio de información**
 - Porcentaje de intercambio de información (**e**)
- **Esquemas con asignación dinámica**
 - Número de subtrabajos a enviar (**t**)

Parámetros, metodología y modelos

- **Esquemas maestro esclavo:**
 - Profundidad del nivel inicial (**I**)
 - Número de procesadores a utilizar (**p**)
- **Esquemas con asignación estática:**
 - Tipo de asignación estática (**A**)
- **Esquemas con intercambio de información**
 - Porcentaje de intercambio de información (**e**)
- **Esquemas con asignación dinámica**
 - Número de subtrabajos a enviar (**t**)

Parámetros, metodología y modelos



Parámetros, metodología y modelos

- **Modelo general de los esquemas paralelos**

$$TE(n, p) = TE_{Secuencial}(l) + TE_{Paralelo}(n - l, p)$$

- Modelo homogéneo parte paralela:

$$TE_{Paralelo}(n - l, p) = \left\lceil \frac{TS(n-l)}{p} \right\rceil * TE_{Secuencial}(n - l)$$

- Modelo asignación parte paralela:

$$TE_{Paralelo}(n - l, p, A) = \max_{j=0}^{p-1} \left\{ \sum_{i=0, A(i)=j}^{NS-1} TE_{Secuencial}(n - l, A(i)) \right\}$$

Parámetros, metodología y modelos

- **Modelo general de los esquemas paralelos**

$$TE(n, p) = TE_{Secuencial}(l) + TE_{Paralelo}(n - l, p)$$

- Modelo homogéneo parte paralela:

$$TE_{Paralelo}(n - l, p) = \left\lceil \frac{TS(n-l)}{p} \right\rceil * TE_{Secuencial}(n - l)$$

- Modelo asignación parte paralela:

$$TE_{Paralelo}(n - l, p, A) = \max_{j=0}^{p-1} \left\{ \sum_{i=0, A(i)=j}^{NS-1} TE_{Secuencial}(n - l, A(i)) \right\}$$

Parámetros, metodología y modelos

- **Modelo general de los esquemas paralelos**

$$TE(n, p) = TE_{Secuencial}(l) + TE_{Paralelo}(n - l, p)$$

- Modelo homogéneo parte paralela:

$$TE_{Paralelo}(n - l, p) = \left\lceil \frac{TS(n-l)}{p} \right\rceil * TE_{Secuencial}(n - l)$$

- Modelo asignación parte paralela:

$$TE_{Paralelo}(n - l, p, A) = \max_{j=0}^{p-1} \left\{ \sum_{i=0, A(i)=j}^{NS-1} TE_{Secuencial}(n - l, A(i)) \right\}$$

Parámetros, metodología y modelos

- **Modelo del esquema M/E asignación estática con intercambio de información**

$$\begin{aligned}
 TE_{Paralelo}(n-l, p, e) &= k_{MedioParalelo} \cdot \left(\frac{NNG(n-l) \cdot TS(n-l)}{p-1} \right) \cdot TCN \\
 &+ TC \cdot \frac{NNG(n-l) \cdot k_{MedioParalelo}}{e}
 \end{aligned}$$

Resultados experimentales (Autoconfiguración de parámetros)

- **Esquema M/E con asignación estática sin intercambio de información**
 - Implementación en memoria compartida (OpenMP)
 - Implementación de los dos modelos teóricos
 - Distribución cíclica, bloques, aleatoria y voraz.
- **Esquema M/E con asignación estática con intercambio de información**
 - Implementación en memoria distribuida (MPI)
 - Implementación de inferencia a partir de datos y búsqueda local

Resultados experimentales - EMEAE sin intercambio

	l	p	A	TR	UserOpt	UserMed	UserLow
<i>ln_40₁</i>	2	4	c	21.26	13.18	21.26	27.27
<i>ln_40₂</i>	2	4	c	0.095	0.07057	0.0949	0.1116
<i>ln_40₃</i>	1	1	c	0.00031	0.00029	0.00043	0.00037
<i>ln_40₄</i>	18	4	v	49.13	67.37	90.78	128.326
<i>ln_40₅</i>	4	4	v	117.56	121.54	150.93	228.02

Resultados experimentales - EMEAE con intercambio

	l	p	e	TR	UserOpt	UserInst	UserLow
<i>ln_40₁</i>	12	14	1000	29.40	7.37	11.64	170.57
<i>ln_40₂</i>	13	10	1000	0.135	0.030	0.033	0.4804
<i>ln_40₃</i>	13	12	1000	0.00031	0.0002	0.0002	0.000234
<i>ln_40₄</i>	16	16	1000	90.67	24.65	54.37	1233.31

Conclusiones

- El modelado de la técnica plantea **problemas** al existir alta **dependencia de las entradas**. Es necesaria incluir información de la entrada para refinar la información obtenida en la instalación.
- El paralelizar este tipo de técnicas plantea problemas de **configuración de los parámetros** para aprovechar el paralelismo.
- Las **metodologías** propuestas obtienen **resultados parcialmente favorables**. Pese a quedar lejos del óptimo, se proponen valores de los parámetros que permiten reducir los tiempos secuenciales.

Trabajos futuros

- **Extender el estudio** de los esquema con intercambio de información **a otro tipo de plataformas**
- Proponer nuevas técnicas para afinar la estimación de los nodos generados y **acercarnos más al tiempo de ejecución**
- Implementar un **esquema secuencial con paralelización interna y autoconfiguración**. Añadirlo al proyecto fin de carrera de Septiembre de 2008: Herramienta web de esquemas algorítmicos secuenciales
- **Aumentar el banco de pruebas** con tamaños más grandes de las entradas y aplicar a un **problema real**
- **Estudiar otras técnicas de recorrido de árboles** (*Branch and bound*)

Técnicas de autooptimización en recorridos de árboles por medio de *backtracking*

Manuel Quesada Martínez

25 de febrero de 2009