

CERCA LOCAL

Marc Asenjo i Ponce de León Joan Marcè i Igual
Iñigo Moreno i Caireta

11 d'abril de 2016

1 Implementació estat

La nostra implementació d'estat utilitza dos vectors de 2 dimensions. Un conté els fitxers que ha demanat cada usuari i l'altre vector, el servidor que serveix aquest fitxer. Així doncs al modificar l'estat només es modifica el segon vector permetent que el primer sigui una variable **static**.

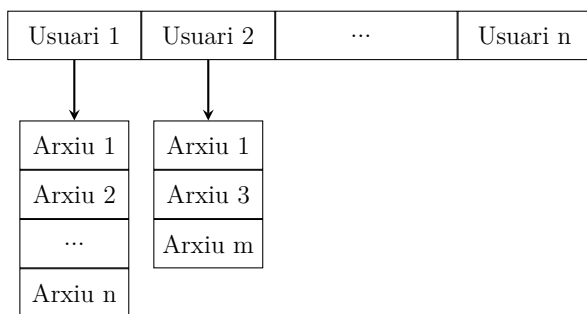


Figura 1: Representació del vector de fitxers sol·licitats per cada petició

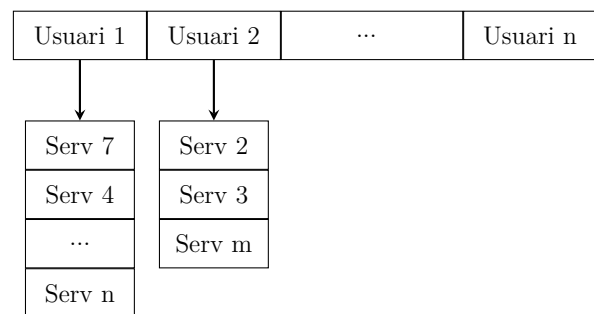


Figura 2: Representació del vector dels servidors assignats a cada petició

2 Operadors

S'han creat dos operadors diferents, tots dos basats en el mateix principi. El primer operador (1) canvia el servidor que serveix una petició. Donats un identificador de usuari (**uid**), una petició (**rid**) i un servidor (**sid**). Assigna el servidor **sid** perquè envii la petició **rid**.

```
public void swapServer(int uid, int rid, int sid);
```

El segon operador (2) és igual que el primer però aplicant-ho dos cops de manera que es canvien els servidors assignats a dues peticions. Això permet generar successors molt diferents els uns dels altres de manera que (en teoria) es poden trobar més solucions tot i que el temps d'execució és més elevat.

3 Estratègies solució inicial

S'han plantejat dues estratègies generadores de la solució inicial, la primera, per cada petició, assigna un servidor aleatori que contingui el fitxer demanat; la segona, assigna a cada petició el servidor amb menys temps de transmissió per aquesta petició.

4 Funcions heurístiques

Tal i com s'explica a l'enunciat s'han de crear dos heurístics, un per tal de minimitzar el temps del servidor que necessita més temps per transmetre les seves peticions i un altre per minimitzar el temps total de transmissió però amb un temps per servidor el més similar possible entre tots els servidors.

Pel primer heurístic s'ha agafat com a valor de l'heurístic el valor de temps del servidor que triga més.

Pel segon heurístic s'ha agafat com a valor de l'heurístic la següent fórmula sobre el conjunt de dades dels temps de transmissió de cada servidor.

$$\mu \cdot \bar{x}^2 + \sigma^2$$

On σ és la variància, \bar{x} és la mitjana i μ és un factor que s'ha d'escollir experimentalment.

5 Experiments

5.1

Es demana escollir el conjunt d'operadors que dona millors resultats per a la funció heurística que optimitzi el primer criteri. Per obtenir els resultats s'han utilitzat 10 llavors diferents, han estat les mateixes 10 tant per l'operador 1 com pel 2. Als resultats es pot veure que les solucions dels operadors 1 i 2 són molt semblants, la solució obtinguda amb l'operador 2 és millor però en canvi el temps per trobar la solució és tant gran que ni tan sols s'aprecia al gràfic pel que s'acaba escollint l'operador 1.

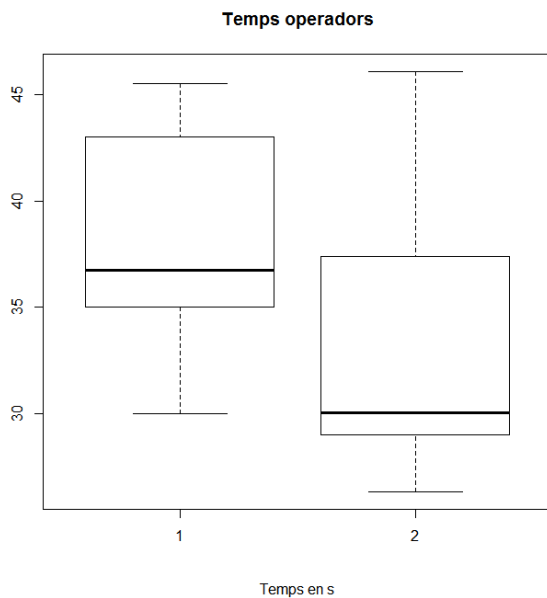


Figura 3: Box plot dels temps màxims dels resultats al utilitzar l'operador 1 i 2

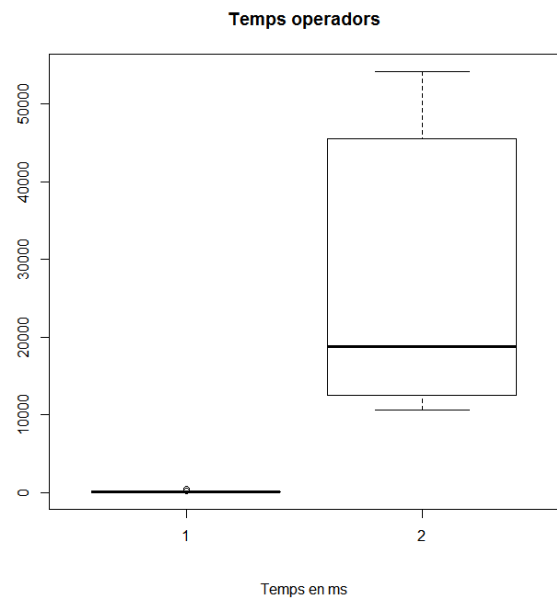


Figura 4: Box plot dels temps que es triga en calcular la solució amb els operadors 1 i 2

Operador 1		Operador 2	
Resultat (s)	Temps (ms)	Resultat (s)	Temps (ms)
35	415	30	45.523
41	110	29	54.225
35	168	37,4	16.391
35,5	104	34	11.272
45,5	81	46,1	10.543
35,5	91	30,1	15.442
43	33	28,4	34.781
30	89	29,9	21.260
38	53	40,3	12.497
44	79	26,3	47.050

Taula 1: Resultats i temps obtinguts amb els operadors 1 i 2

5.2

Es demana comparar l'estratègia de generació de la solució inicial amb els criteris de l'apartat anterior. També s'han utilitzat 10 llavors diferents per obtenir els paràmetres inicials.

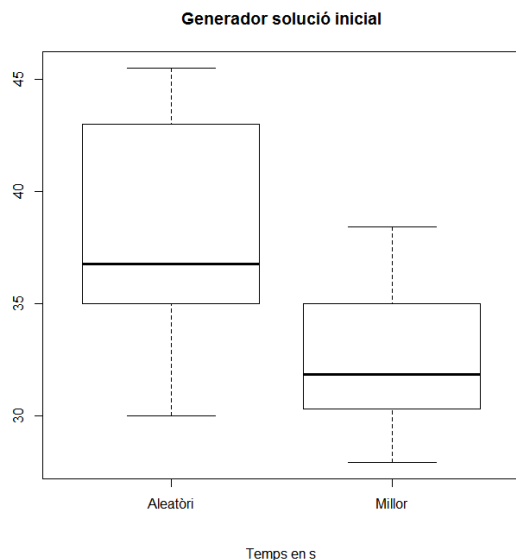


Figura 5: Box plot dels temps màxims dels resultats

Aleatori	Millor servidor
35	32,8
41	35,6
35	32,4
25,5	30,3
45,5	38,4
35,5	29,7
43	35
30	27,9
38	30,6
44	31,3

Taula 2: Resultats (en segons) dels dos algoritmes de generació

5.3

Es demana determinar els paràmetres del Simmulated Annealing amb

		λ		
		0,001	0,01	0,1
k	1	25.610	25.497	30.733
	1e2	25.867	25.577	30.788
	1e4	26.666	25.715	30.474

6 Comparació resultats