Objetivos

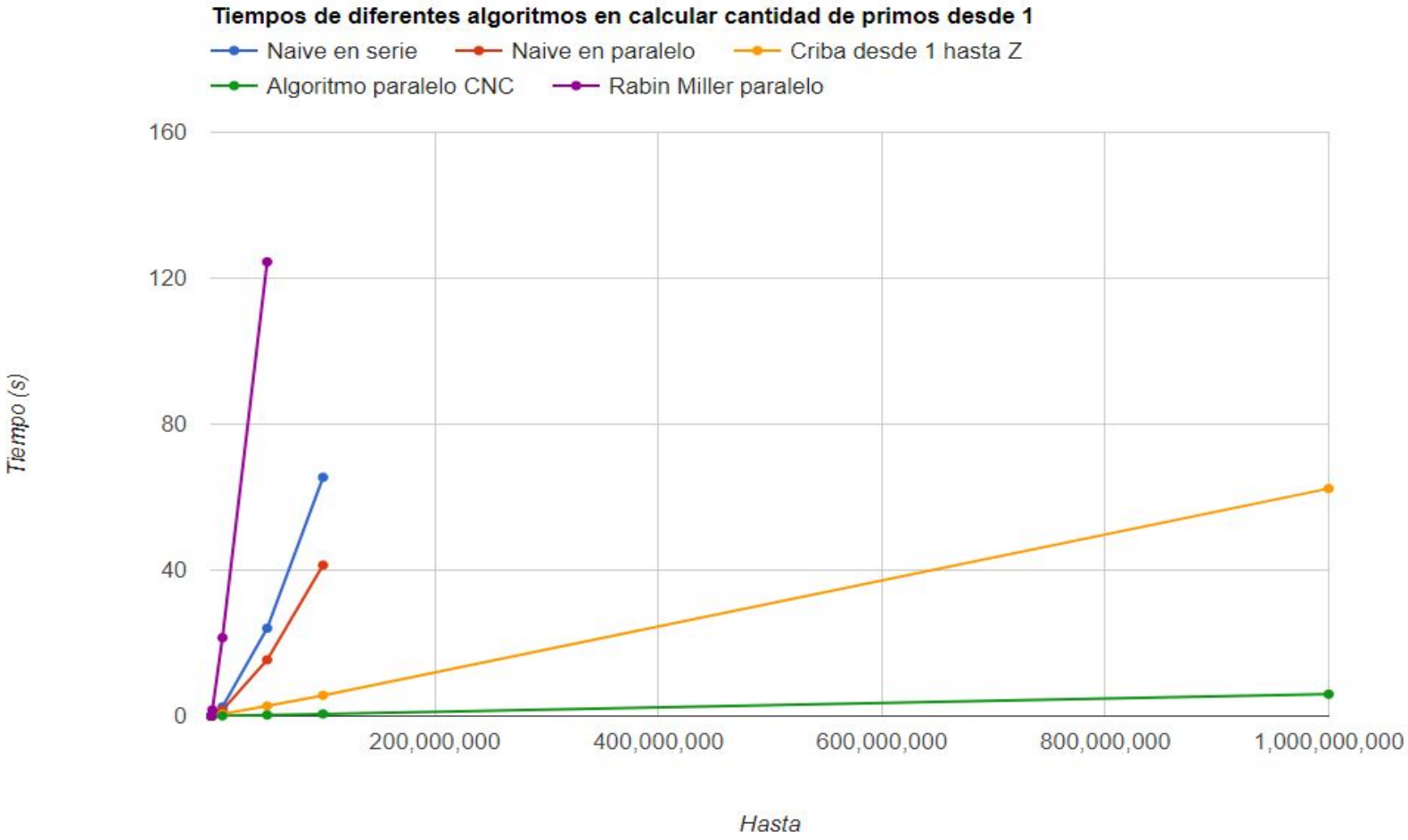
- Estudiar el comportamiento del algoritmo de criba en rango variando el rango que procesa cada consumer
- Implementar una solución cnc que aplique el test probabilístico rabin miller para cada número en paralelo (para el rango que usamos, es determinístico)
- Comparar esta nueva solución con la que nos había dado mejores resultados: la criba en rango

Variando el rango

Rango/ ChunkSize	1k	10k	25k	50k	100k	250k	500k	10m
1-1000 m	12.78	6.35	5.80	5.78	5.31	5.34	5.52	5.9
1-10m	0.07074	0.0647	0.0642	0.0656	0.0672	0.0770	0.0865	0.1025

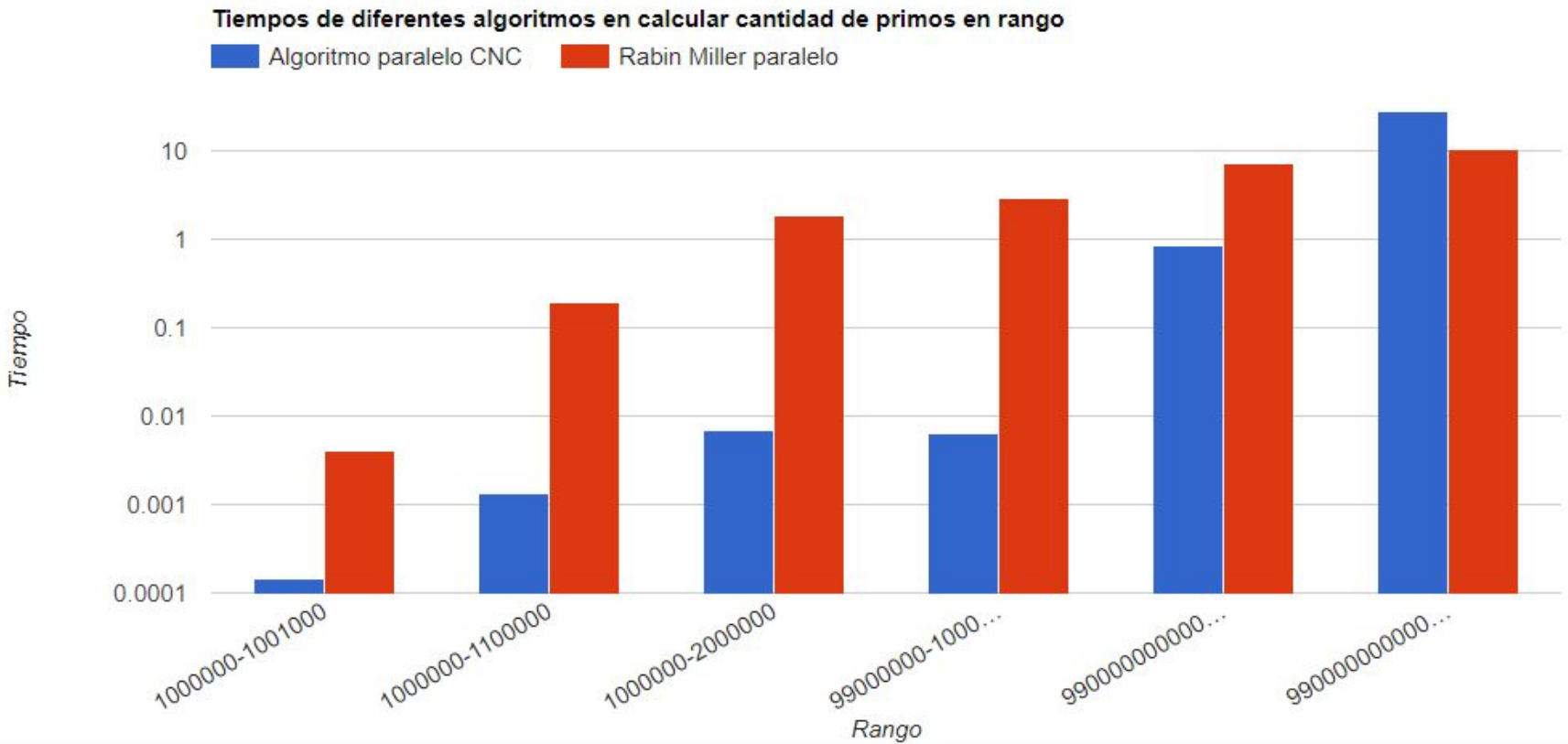
Conclusiones

- Puede observarse que la performance varía de acuerdo al rango procesado en cada caso, con un comportamiento que va mejorando al aumentar el rango hasta llegar a cierto punto, a partir del cual va empeorando ya que se acerca a perder el paralelismo que había
- En el primer caso el mejor rango fue el que generó 10000 procesos para los consumers, mientras que en el segundo caso fue el que generó 400 procesos
- Dependiendo del rango estudiado lo óptimo pareciera variar, elegimos el rango de a 10000 como parámetro por defecto dado que fue razonable en ambos casos


Tiempos de diferentes algoritmos en calcular cantidad de primos desde 1
Naive en serie
Naive en paralelo
Criba desde 1 hasta Z
Algoritmo paralelo CNC
Rabin Miller paralelo
160
120
80
40
0
Tiempo (s)
200,000,000
400,000,000
600,000,000
800,000,000
1,000,000,000
Hasta

Análisis

- Rabin miller es muy costoso, para números chicos es mejor la solución de ir hasta la raíz del número. Recordemos que la complejidad de rabin miller es O(k log^3 n). Incluso considerando el peor caso testeado de 100 millones, la raiz es 10000 mientras que con rabin miller realizamos al 18768 operaciones por cada testigo (usamos 9 testigos).
- El número debe ser muy grande para que rabin miller sea mejor que ir hasta la raíz

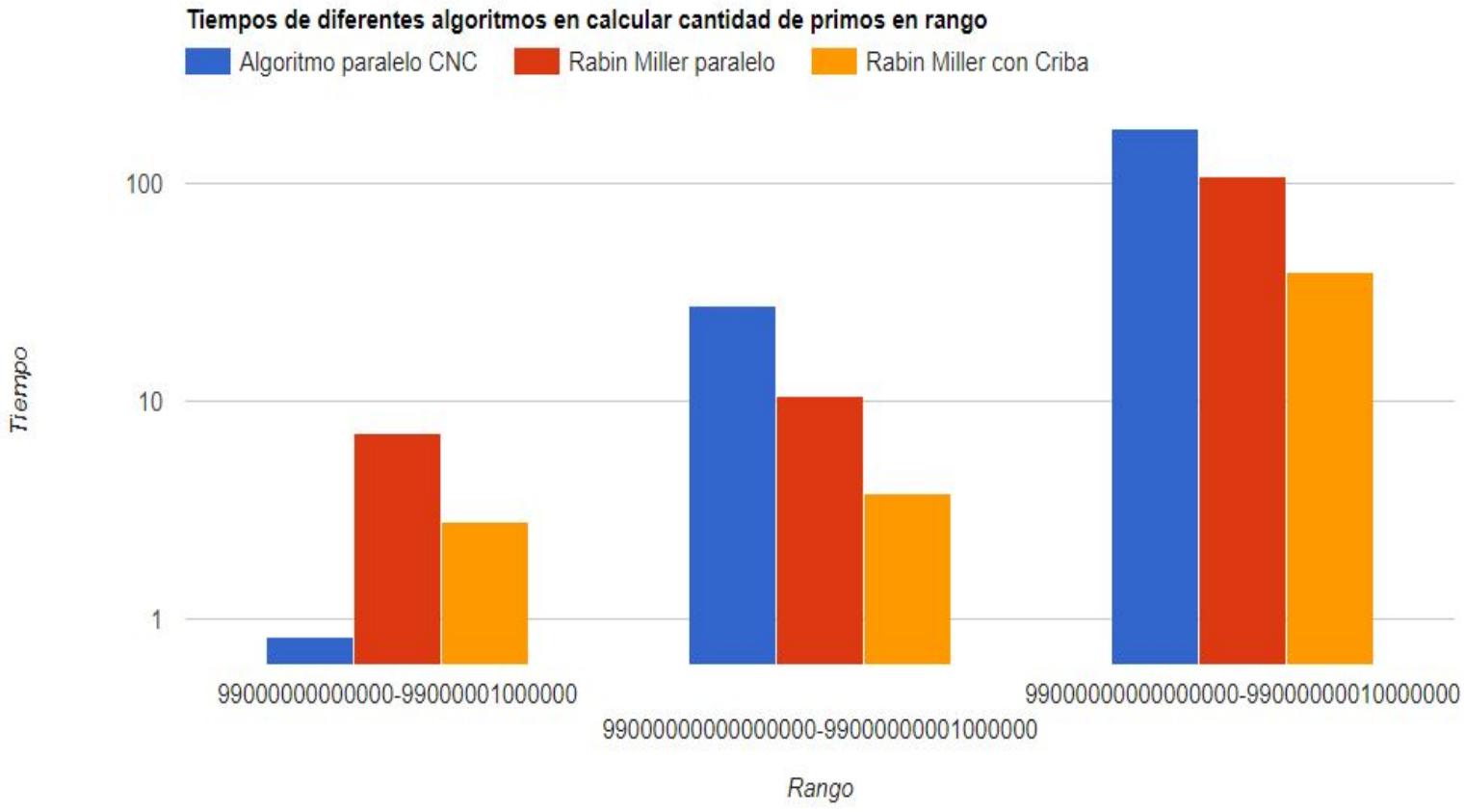

Tiempos de diferentes algoritmos en calcular cantidad de primos en rango
Algoritmo paralelo CNC
Rabin Miller paralelo
10
1
0.1
0.01
0.001
0.0001
Tiempo
1000000-1001000
1000000-1100000
1000000-2000000
99000000-1000…
990000000000…
990000000000…
Rango

Análisis

- Cuanto más grande son los números con los que trabajamos, más se puede apreciar la performance de rabin miller. Cuando el rango que se quiere trabajar es pequeño pero los números son grandes, rabin miller pareciera mejorar la mejor solución que teníamos de criba en rango.
- Rabin miller es costoso por cada número, pero nos evita tener que calcular los primos hasta la raíz del rango con el que queremos trabajar. Dependiendo de con cuántos números trabajamos y en qué rango, puede convenir uno u otro algorimo

Propuesta

- Hacer algo que una los beneficios de ambos: bajo costo por número (criba) pero poder trabajar con números grandes sin demasiado overhead causado por los primos hasta la raíz (rabin miller)
- Para esto hicimos un nuevo programa que combina ambas ideas: hace la criba igual que antes, pero en lugar de ir calcular los primos hasta la raíz del rango que necesitamos, se limita a cierto límite (seteado en 500000).
- Si la criba nos dice que un número no es primo, no lo es
- Si nos dice que sí lo es, puede ser un falso positivo ya que no tenemos todos los primos hasta la raíz. En estos casos aplicamos rabin miller para confirmar


Tiempos de diferentes algoritmos en calcular cantidad de primos en rango
Algoritmo paralelo CNC
Rabin Miller paralelo
Rabin Miller con Criba
100
10
1
Tiempo
99000000000000-99000001000000
99000000000000000-99000000001000000
990000000000000000-990000000010000000
Rango

Análisis

- Al trabajar con números cada vez mayores, como habíamos dicho la criba va perdiendo contra el rabin miller que teníamos anteriormente
- El nuevo algoritmo que combina ambas ideas mejora notablemente la performance, en el último caso la combinación de ambos tardó 36 segundos versus los 100 segundos de sólo usando rabin miller y los 180 de sólo criba
- Hemos logrado un buen balance entre ambas ideas, superando el mejor algoritmo que teníamos anteriormente

A futuro

- Implementar los algoritmos con pcr en lugar de cnc puro y estudiar su performance en una máquina con 16 cores