UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO
DE INFORMÁTICA

Rendimiento Operaciones Rank y Select para distintas estructuras de datos con BitVectors

Jean-Pierre Villacura

Compresión de Texto

Diego Arroyuelo

Primer semestre de 2023

Índice

1. Introducción

El siguiente informe discute los resultados obtenidos de la aplicación de distintas estructuras de datos para comprimir BitVectors, considerando soporte para las operaciones de Rank y select, cuando corresponda.

Concretamente, se utilizan los siguientes vectores en las pruebas: **la_vector**, zombit, oz-vector, SD y rrr con distintos parámetros.

Se realiza una comparación en el rendimiento de las estructuras de datos anteriores realizando operaciones de rank y select utilizando archivos que contienen diccionarios y gran cantidad de información de páginas web.

2. Metodología y Resultados

2.1. Metodología

Se utilizan las siguientes librerias para la implementación de estructuras de datos las cuales se busca comprimir datos incluyendo el soporte para las operaciones de **rank** y **select**:

- **sdsl Lite**: Incluye gran cantidad de Succint data Structures que permiten en la práctica ser implementadas de manera simple y de forma intuitiva.
- **la_vector**: Bitvector/Container comprimido que soporta queries de rank y select de forma eficiente utilizando novedosas formas de compresión.
- **s18_vector**: Add-on para la libreria Sdsl-Lite que implementa el vector del mismo nombre en forma de bit vector compressed. Soporta operaciones de rank y select, propuestas por Arroyuelo (2018)
- **zombit_vector**: Otro Add-on propuesto por Adrián Gomez para la libreria Sdsl-Lite.

Concretamente, los parámetros utilizados son los siguientes:

- **oz_vector:** rank_support_oz<1> , select_support_oz<1>.
- **zombit_vector:** rank_support_zombit<1>.
- **sd:** select_support_sd<1>, rank_support_sd<1>.
- **s18:** rank_support<1,X>, select_support<1,X>. $x \in [8, 16, 32, 64, 128, 256]$.
- **rrr_vector:** rank_support_rrr<1,Y>, select_support_rrr<1,Y>. $Y \in [31, 63, 127]$.
- **la_vector:** la_vector<uint32_t,Z>. $Z \in [3, 6, 10, 15, 18]$.

3. Discusión de resultados

Se tabulan los resultados obtenidos aplicando las estructuras de datos anteriores a los archivos proporcionados:**BV_coreutils.dat** y **BV_worldleaders.dat** en forma de gráficos que se muestran a continuación:

Otros archivos como BV_einstein_de.dat y BV_einstein_en.dat no fueron considerados al generar error de Segmentation Fault al correr el código para realizar pruebas de rendimiento sobre estos.

3.1. Rank BV-coreutils.dat

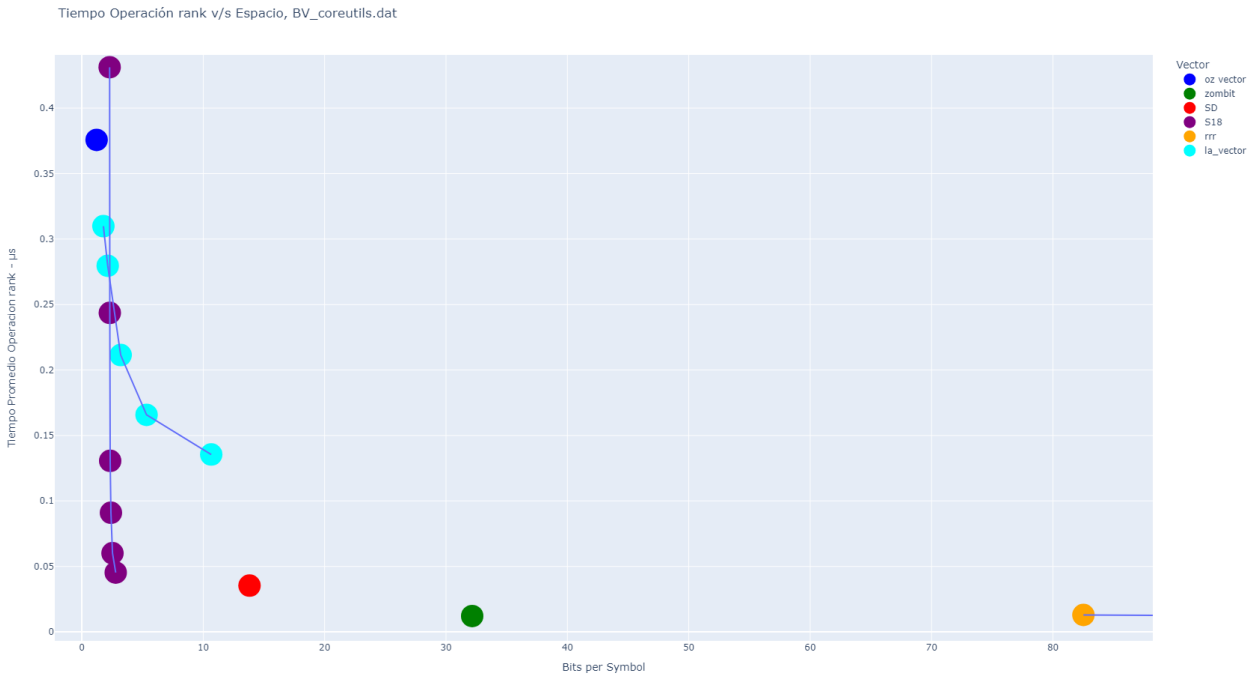

Figura 1: Rendimiento Operación Rank en microsegundos, archivo: BV-coreutils.dat - Vista General (Fuente Propia)

En este gráfico correspondiente a la operación rank del archivo BV-coreutils.dat se observa que las estructuras con menor tiempo de ejecución corresponden a Zombit, SD , rrr y parámetros de la estructura S18.

Con respecto a la-vector, se nota que es más rápida que oz-vector pero significativamente más lenta que gran cantidad de parámetros de la curva S18 y los otros vectores mencionados anteriormente. En cuanto al Bits per Symbol, oz-vector, s18, la-vector y SD son las estructuras cuya magnitud es menor. rrr por otra parte es la estructura que más Bits per Symbol tiene situandose relativamente lejos de las estructuras anteriores.

3.2. Select BV-coreutils.dat

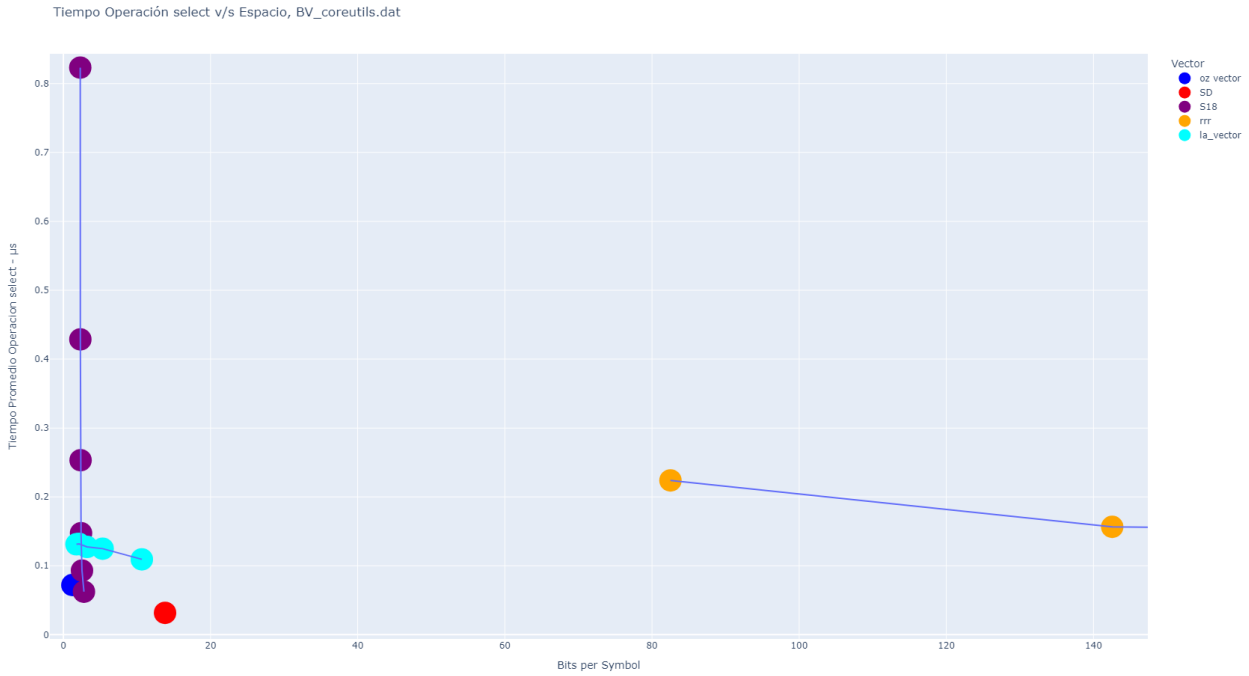

Figura 2: Rendimiento Operación Select en microsegundos, archivo: BV-coreutils.dat - Vista General (Fuente Propia)

En esta figura se muestra una big picture del rendimiento de las estructuras anteriores con la operación Select. Se puede notar que la estructura **rrr** sigue teniendo un valor de Bits per symbol alto y su tiempo de ejecución es menor que otras estructuras. Debido a que gran cantidad de estructuras se concentran en la zona de bajo tiempo de ejecución y bajo Bits per Symbol, se realiza un zoom sobre esta zona para poder observar mejor lo que sucede en esta zona, que se puede ver en la siguiente figura:

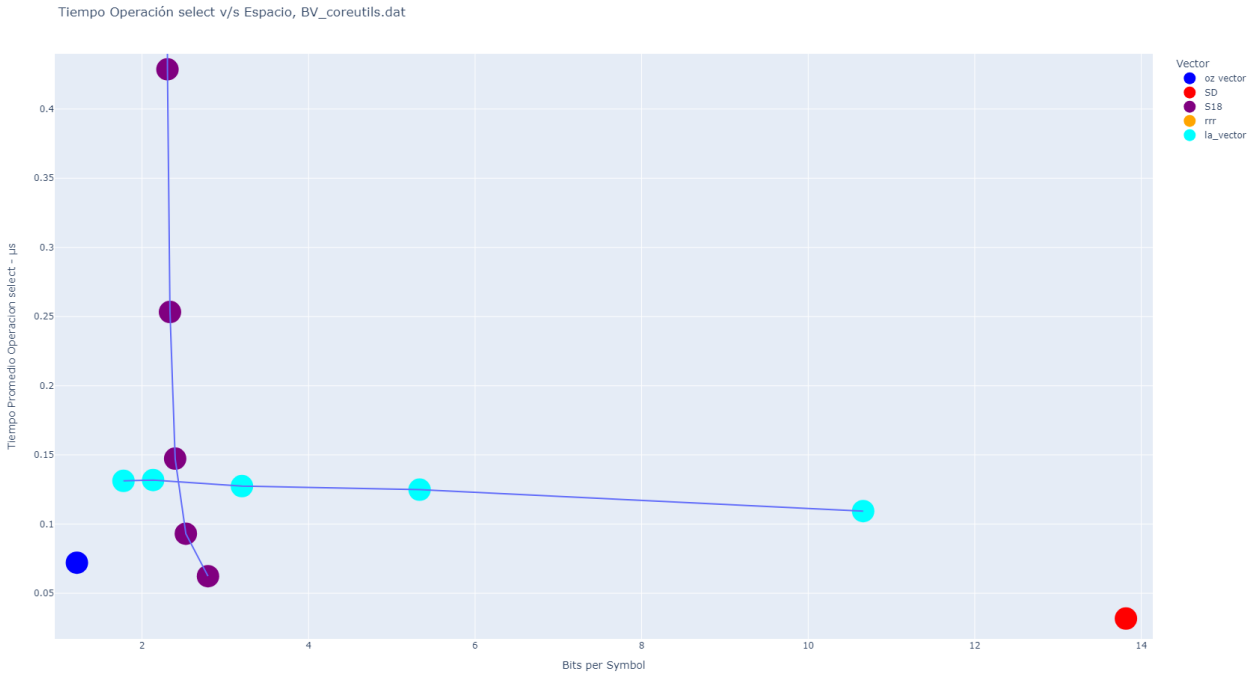

Figura 3: Rendimiento Operación Select en microsegundos, archivo: BV-coreutils.dat - Vista Particular (Fuente Propia)

Con la figura anterior se observa que SD es la estructura que menor tiempo de ejecución tiene, otras estructuras un poco menos rápidas para este archivo fueron oz-vector y algunos parámetros de S18. Con respecto a la-vector, su tiempo de ejecución disminuye ligeramente al variar sus parámetros de ejecución y no destaca por su tiempo de ejecución, sólo siendo más rápida que algunos parámetros de S18 y rrr.

3.3. Rank BV-worldleaders.dat

En adelante se analiza el diccionario BV-worldleaders.dat

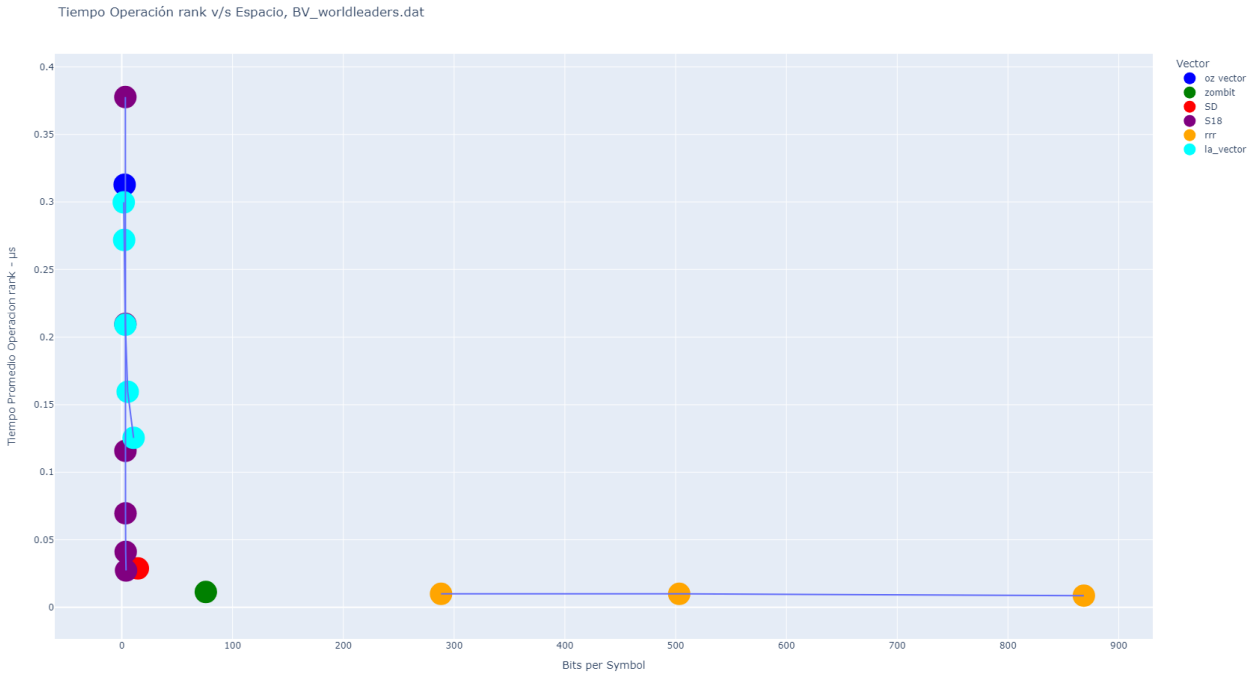

Figura 4: Rendimiento Operación Rank en microsegundos, archivo: BV-worldleaders.dat - Vista General (Fuente Propia)

Para la operación rank, se observa que zombit y la curva de rrr tienen el menor tiempo de ejecución, seguido popr SD y parte de la curva de S18. En cuanto al Bits per Symbol, se observa que rrr mantiene su comportamiento de 'alejarse' de las otras estructuras de datos al tener un valor relativamente alto. En general el Bits per symbol se mantiene para rrr, oz-vector, zombit, sd y la-vector respecto al archivo anterior.

Dado que para este informe resulta relevante la-vector, se realiza un zoom sobre su curva con el objetivo de analizar su comportamiento.

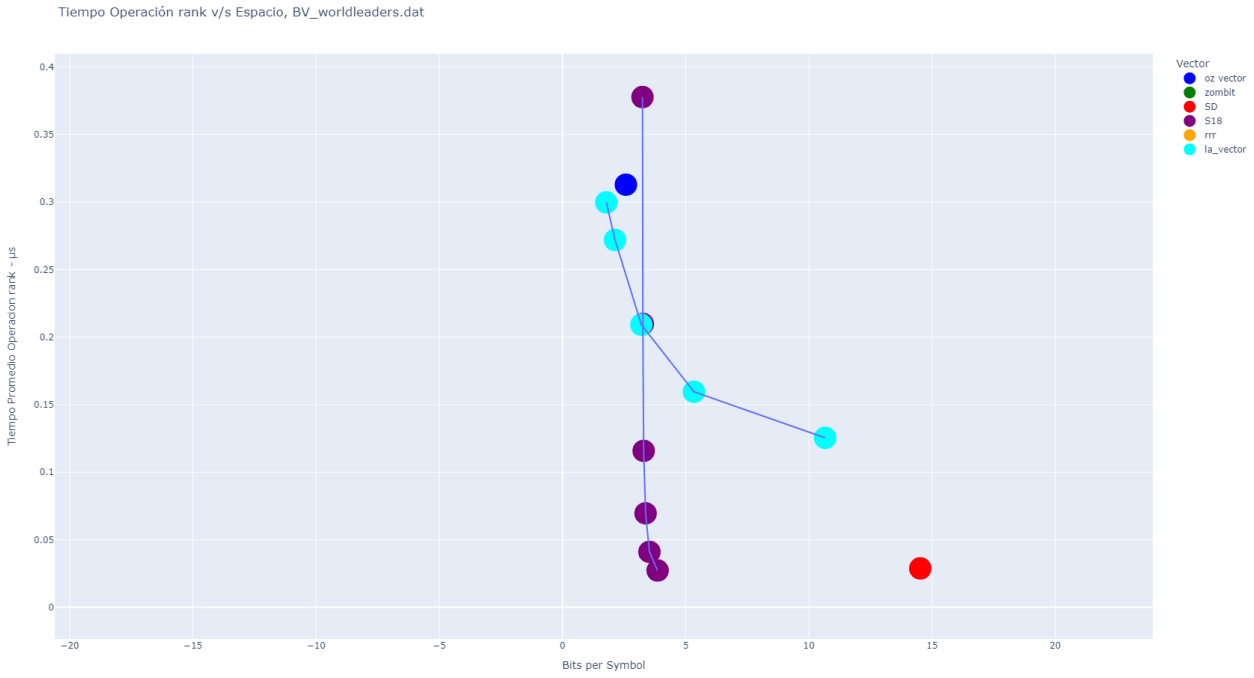

Figura 5: Rendimiento Operación Rank en microsegundos, archivo: BV-worldleaders.dat - Vista Particular (Fuente Propia)

Realizando un zoom sobre la zona de la-vector, sin considerar a las estructuras zombit y rrr que tienen un tiempo de ejecución aún menor, se observa que el comportamiento de la-vector respecto a sus parámetros cambia considerablemente el tiempo de la operación rank. Sin embargo sigue siendo precedido por algunos parámetros de la curva S18 que alcanza a ser tan rápida como SD y más lenta que Zombit y rrr.

3.4. Select BV-worldleaders.dat

Utilizando la operación Select, se obtiene el siguiente gráfico:

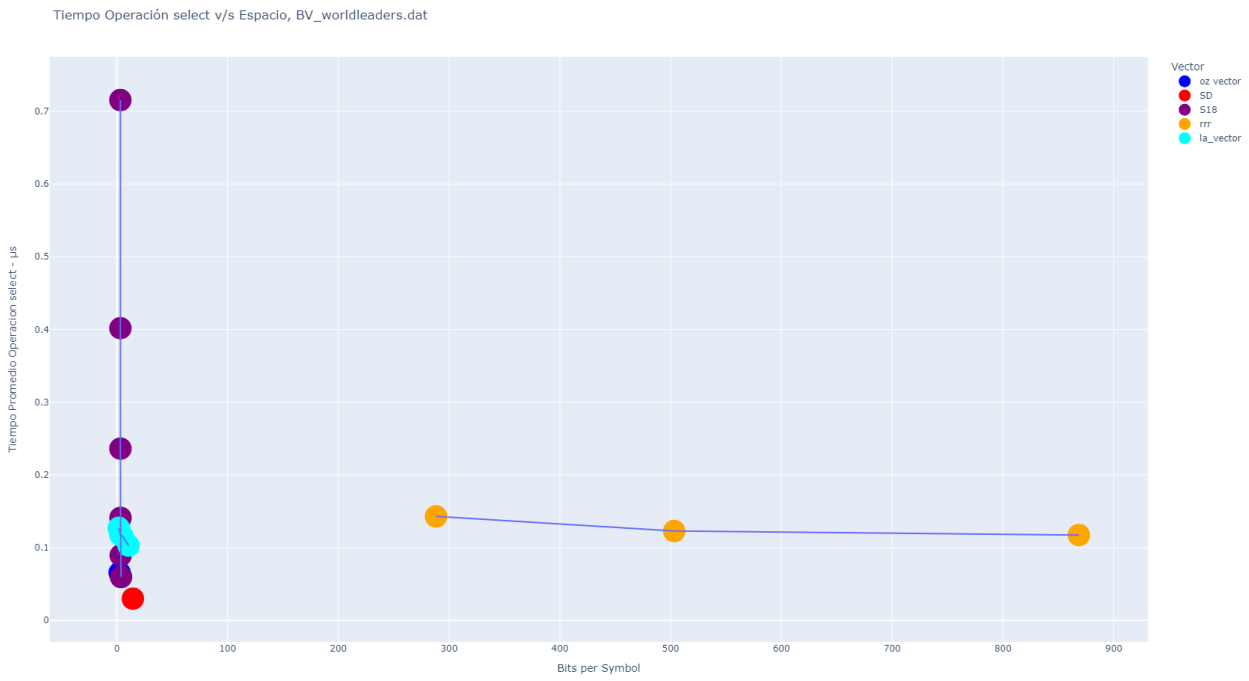

Figura 6: Rendimiento Operación Select en microsegundos, archivo: BV-worldleaders.dat - Vista General (Fuente Propia)

Con una visión big picture, se puede notar que rrr deja de ser la estructura con un tiempo de ejecución pequeño, oz-vector SD y determinados parámetros de la curva S18 conforman las estructuras con menor tiempo de ejecución para la operación select. En concreto se requiere hacer un zoom sobre la zona de bajo Tiempo de ejecución - bajo Bits per Symbol para analizar de mejor manera el comportamiento de las estructuras en determinada zona:

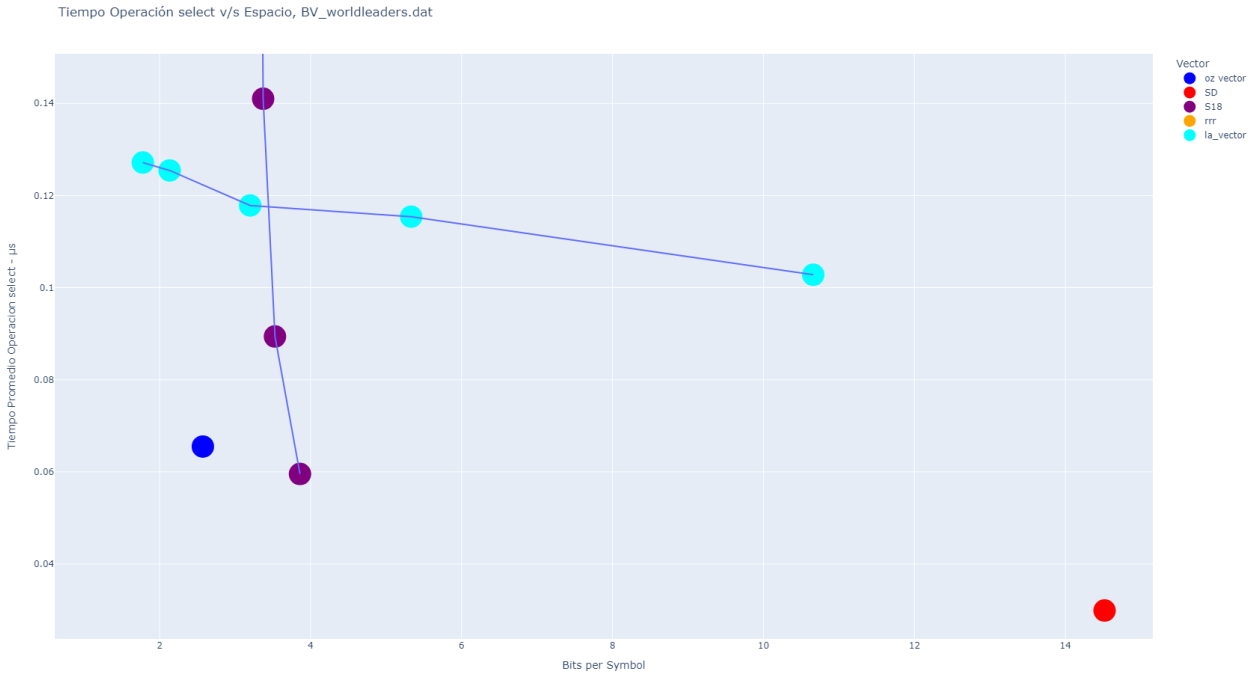

Figura 7: Rendimiento Operación Select en microsegundos, archivo: BV-worldleaders.dat - Vista Particular (Fuente Propia)

Al realizar un zoom sobre la zona, dejando de considerar a la estructura rrr , se observa que la-vector tiene un comportamiento parecido en la curva de tiempo de ejecución en la operación select respecto a la de rank y se nota que significativamente la variación de los parámetros disminuye el tiempo de ejecución, sin alcanzar la rapidez que tienen otras estructuras como S18 y oz-vector.

4. Conclusiones

Tomando en consideración los resultados obtenidos, se concluye:

- En la operación rank, **rrr** es la estructura con menor tiempo de operación asociado, seguido por **zombit** y **SD** las cuales tienen un tiempo de ejecución ligeramente menor.

  Es importante escoger el parámetro adecuado para **S18** y **la-vector** ya que implica que el tiempo de operación empleado varíe significativamente. Este comportamiento es más pronunciado para la curva **S18**, donde en las gráficas es posible notar que una elección no apropiada de S18 puede convertir a la estructura como una de las menos competitivas entre las evaluadas ó bien sea el caso una de las mejores.

  Oz-vector se mantuvo como una de las estructuras con mayor tiempo de ejecución asociado y menor bits per symbol.

  Las estructuras con menos Bits per symbol en los resultados son la-vector, oz-vector, S18, estando en el otro extremo rrr y zombit entre medio de ambas.

- En la operación select, **SD** destaca por su menor tiempo de ejecución, seguido por determinados parámetros de la curva **S18** y oz-vector. **rrr** se sitúa con un mayor tiempo de ejecución estando cercano al de **la-vector**

  La cantidad de Bits per symbol en las distintas estructuras de datos en la operación select sigue manteniendo las mismas proporciones que en la de la estructura rank.

- Se concluye a nivel general que la estructura de **la-vector** en las operaciones de rank y select tiene un rendimiento similar a las estructuras evaluadas en este informe, con un Bits per Symbol entre 0 y 12, valores cercanos al de otras estructuras como S18 y SD.

5. Referencias

Arroyuelo, D. e. a. (2018). Hybrid compression of inverted lists for reordered document collections. information processing and management..