Universidad Nacional de San Agustin

Ciencia de la Computación

Alto rendimiento en coincidencia de patrones en plataformas heterogéneas

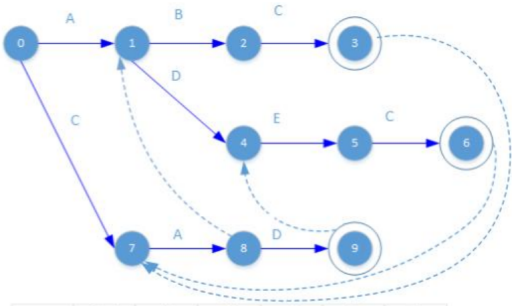
Alumnas :
Judith Escalante Calcina
Rosa Paccotacya Yanque
Profesor:
Mg.Alvaro Henry Mamani Aliaga

Índice

1. Introducción

El descubrimiento de patrones es una de las tareas fundamentales en bioinformática y el reconocimiento de patrones esuna técnica poderosa para busacr coincidencias de patrones en bases de datos se secuencias biológicas. Los algoritmos rápidos y de alto rendimiento son muy demandados en muchas aplicaciones en bioinformática y biología molecular computacional ya que el aumento significativo en el número de secuencias de ADN y proteínas, amplía la necesidad de elevar el rendimiento de los algoritmos de correspondencia de patrones. En este trabajo presentamos la replica de una implementación eficiente de Aho-Corasick (AC) que es un algoritmo de coincidencia de patrones exactos bien conocido con complejidad lineal, y el algoritmo Aho-Corasick (PFAC) Paralelo sin Fallos que es la versión masivamente paralelizada del algoritmo AC sin transiciones de falla , en una arquitectura CPU / GPU heterogénea. Progresivamente rediseñamos los algoritmos y estructuras de datos para que se ajusten a la arquitectura de la GPU. Nuestros resultados en diferentes conjuntos de datos de secuencias de proteínas muestran que la nueva aplicación se ejecuta 15 veces más rápido en comparación con la aplicación original del algoritmo PFAC.

2. Aho Corasick



	Final	A	B	C	D	E
0	0	1	0	7	0	0
1	0	0	2	0	4	0
2	0	0	0	3	0	0
3	1	7	7	7	7	7
4	0	0	0	0	0	5
5	0	0	0	6	0	0
6	1	7	7	7	7	7
7	0	8	0	0	0	0
8	0	1	1	1	9	1
9	1	4	4	4	4	4

Figura 1: Algoritmo Aho-Corasick

Cuadro 1: Información de las bases de datos

	Nodos	Lineas	Peso del archivo
Human	36324438	59846	36,4 MB
Alpaca	20293078	33209	20,3 MB
Anolelizard	22664306	34827	22,7 MB
Armadillo	20882544	38202	20,9 MB
Zebrafish	31797807	46452	31,8 MB
Mouse	18204103	45438	18,2 MB

3. Base de datos

Nuestra base de datos son secuencias de proteínas FASTA de diversos animales

- Humano Human 36,4 MB
- Alpaca Alpaca 20,3 MB
- Lagartija Anolelizard 22,7 MB
- Armadillo Armadillo 20,9 MB
- Pez zebra Zebrafish 31,8 MB
- Ratón Mouse 18,2 MB

4. Resultados de experimentos del paper

La base de datos de los autores son secuencias de proteínas FASTA de diversos animales con diferentes pesos.

- Humano Human 26.635 MB
- Alpaca Alpaca 28.057 MB
- Lagartija Anolelizard 22.59 MB
- Armadillo Armadillo 28.914 MB
- Pez zebra Zebrafish 59.714 MB
- Ratón Mouse 31.04 MB

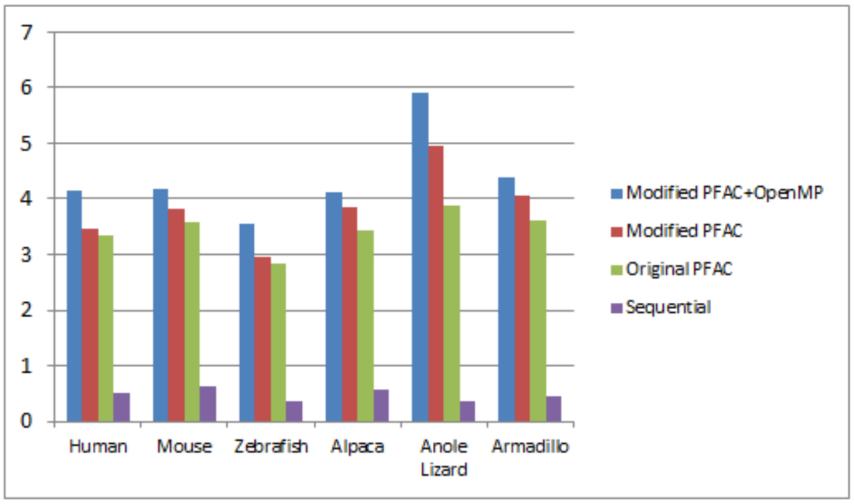


Figura 3: Rendimiento del sistema

	Modified PFAC + OpenMP	Modified PFAC	Original PFAC	Sequential
DB1	6.23	7.08	7.94	53.06
DB2	6.42	6.9	7.83	55.02
DB3	6.05	7.02	7.95	60.04
DB4	6.91	7.12	8.43	49.65
DB5	9.36	11.88	15.42	107.05
DB6	6.7	7.32	8.62	67.32

Figura 2: Tiempo de ejecución (segundos) con todas las bases de datos

5. Resultados de experimentos

Cuadro 2: Tiempo de ejecución (segundos) con todas las bases de datos

	PFAC + openmp	Modified PFAC	Original PFAC	Sequential
Human	10.02	12.23	29.27	123.13
Alpaca	5.47	6.89	11.42	79.05
Anolelizard	6.74	9.73	15.36	85.24
Armadillo	6.21	9.04	12.42	80.34
Zebrafish	7.23	10.75	21.36	94.53
Mouse	5.03	6.78	10.57	70.09

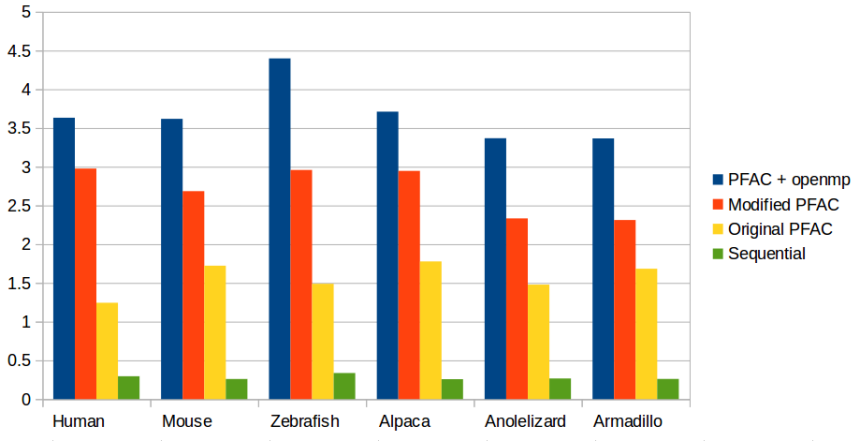


Figura 4: Rendimiento del sistema

Human

```
------------PROGRAMA EN FORMA SERIAL------------
Tiempo del programa serial = 123.134 segundos
------------PROGRAMA EN FORMA PARALELA------------
Tiempo del programa paralelo = 10.0247 segundos
```

Figura 5: Ejecución del código

Figura 6: Monitor del sistema durante ejecución

Alpaca

```
--------------PROGRAMA EN FORMA SERIAL--------------
Tiempo del programa serial = 79.0589 segundos
--------------PROGRAMA EN FORMA PARALELA--------------
Tiempo del programa paralelo = 5.46541 segundos
```

Figura 7: Ejecución del código

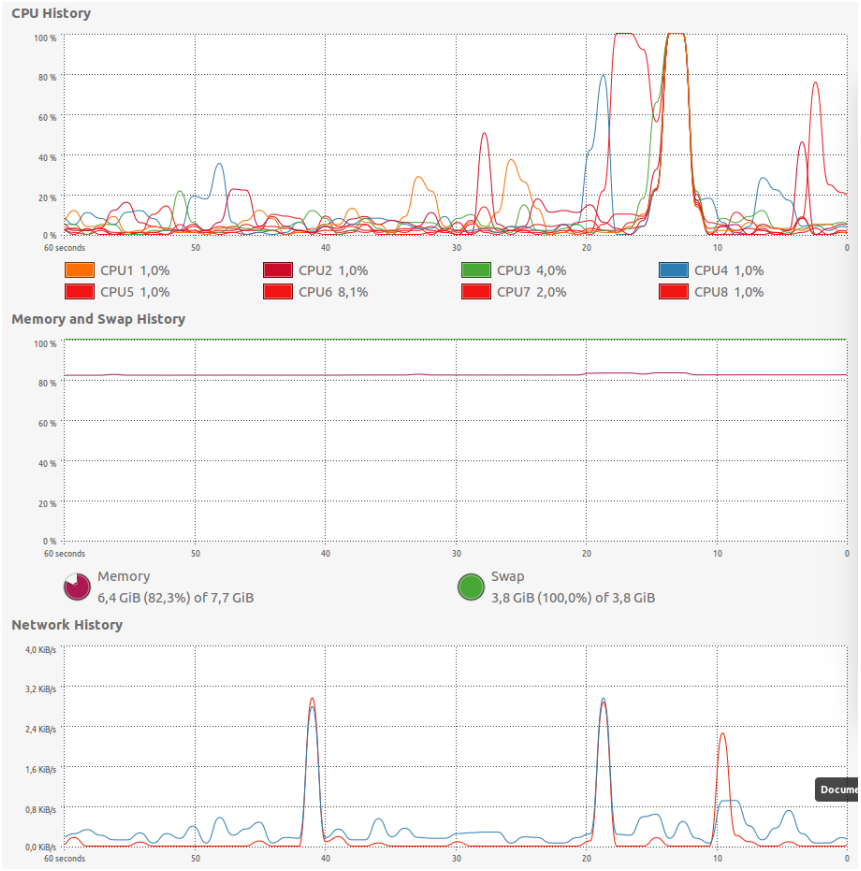


Figura 8: Monitor del sistema durante ejecución

Anolelizard

```
------------PROGRAMA EN FORMA SERIAL------------
Tiempo del programa serial = 85.2453 segundos
------------PROGRAMA EN FORMA PARALELA------------
Tiempo del programa paralelo = 6.73546 segundos
```

Figura 9: Ejecución del código

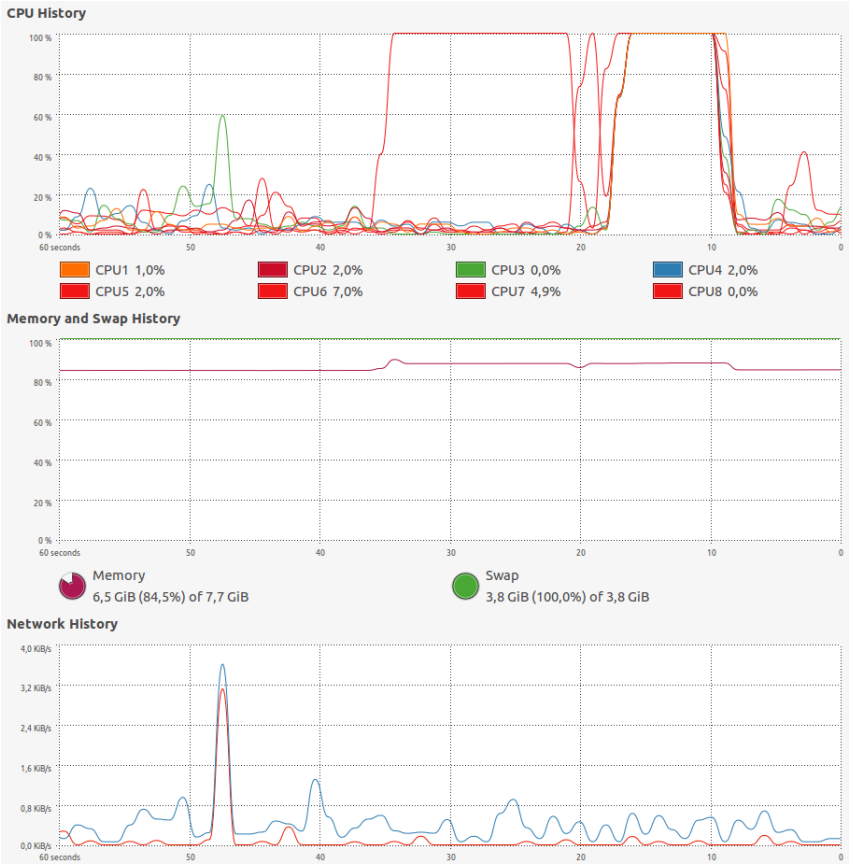


Figura 10: Monitor del sistema durante ejecución

Armadillo

```
------------PROGRAMA EN FORMA SERIAL------------
Tiempo del programa serial = 80.3414 segundos
-------------PROGRAMA EN FORMA PARALELA-------------
Tiempo del programa paralelo = 6.205 segundos
```

Figura 11: Ejecución del código

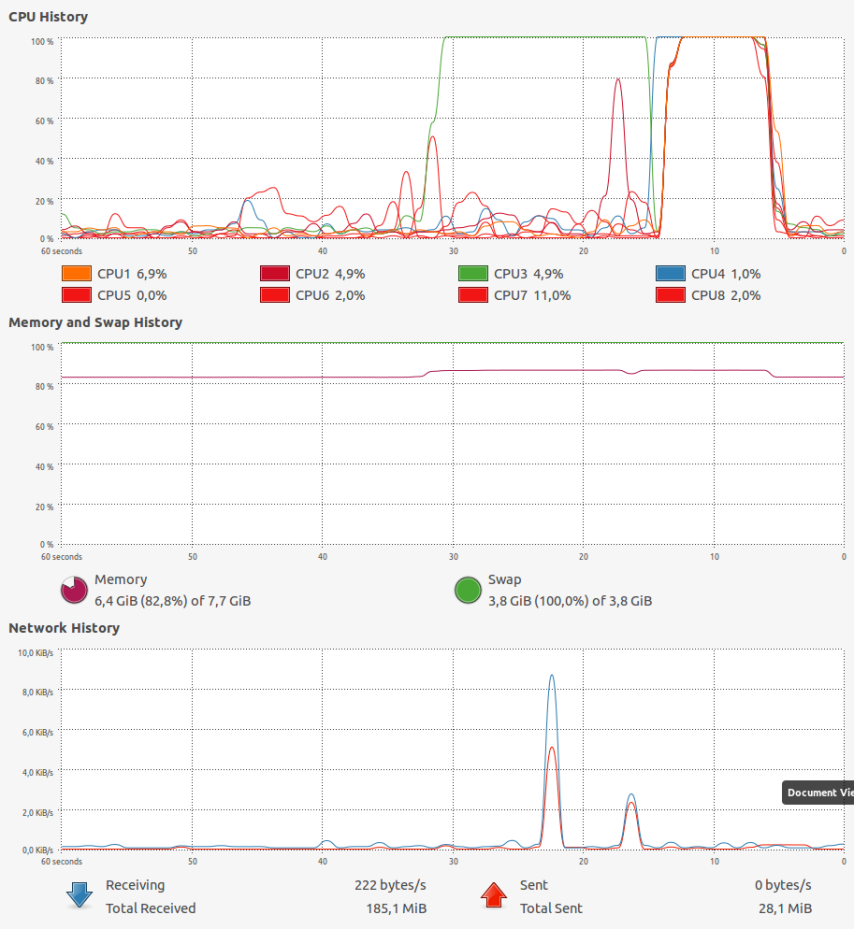


Figura 12: Monitor del sistema durante ejecución

Zebrafish

```
------------PROGRAMA EN FORMA SERIAL------------
Tiempo del programa serial = 94.5353 segundos
------------PROGRAMA EN FORMA PARALELA------------
Tiempo del programa paralelo = 7.2285 segundos
```

Figura 13: Ejecución del código

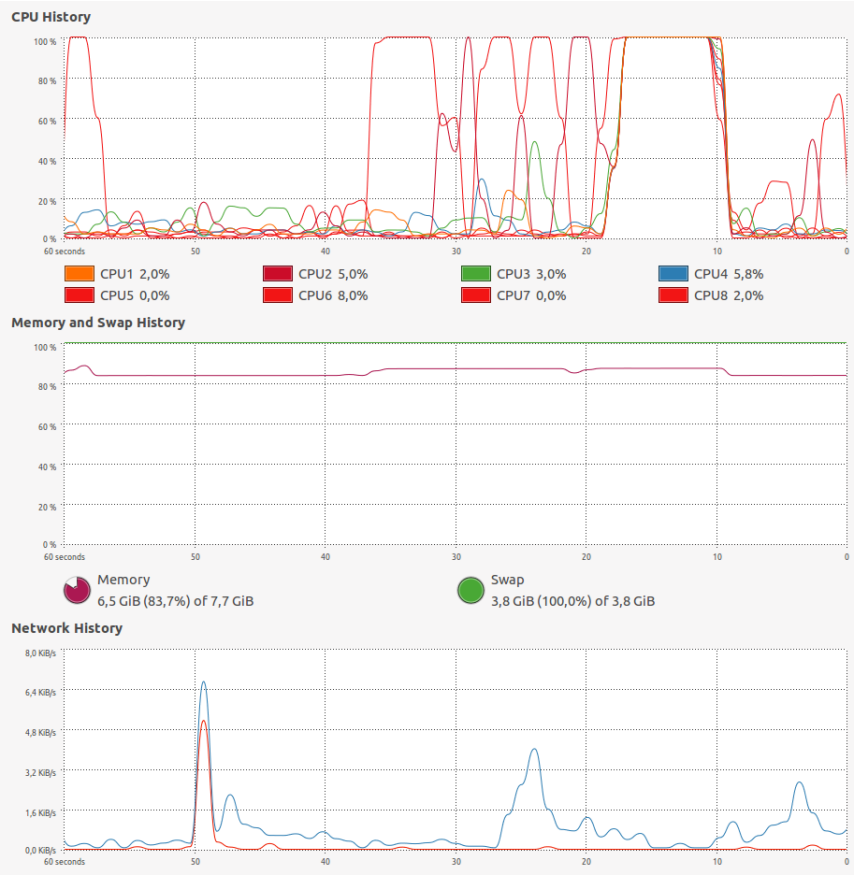


Figura 14: Monitor del sistema durante ejecución

Mouse

```
------------PROGRAMA EN FORMA SERIAL------------
Tiempo del programa serial = 70.0924 segundos
------------PROGRAMA EN FORMA PARALELA------------
Tiempo del programa paralelo = 5.0312 segundos
```

Figura 15: Ejecución del código

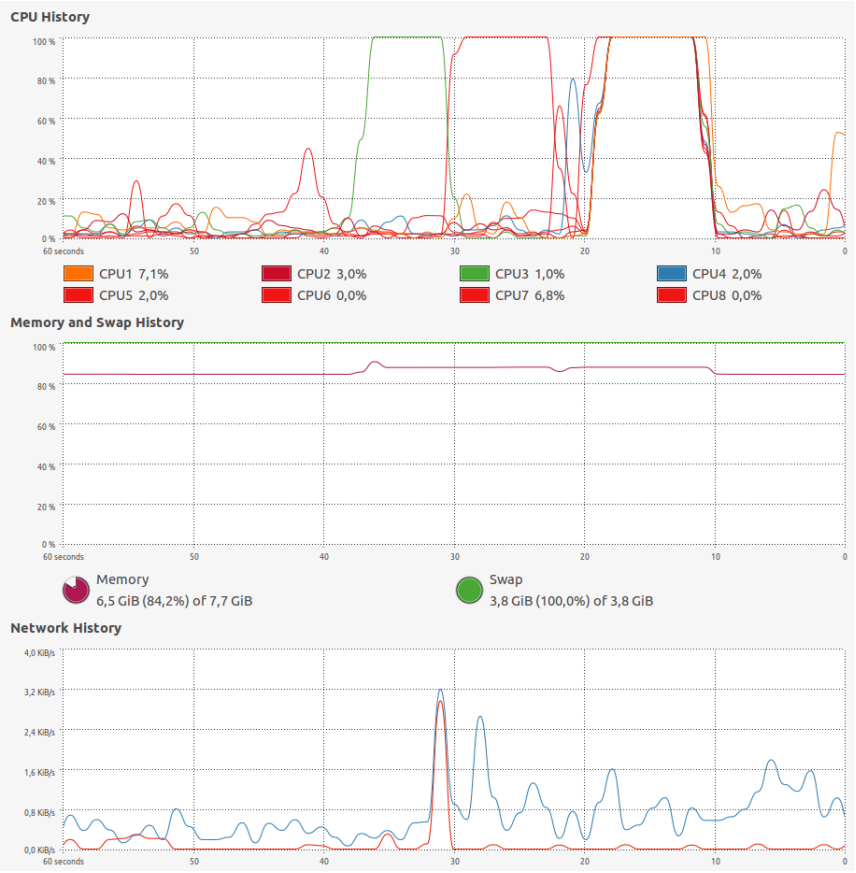


Figura 16: Monitor del sistema durante ejecución