Generación de recomendaciones de ítems musicales basado en las valoraciones implícitas y las similitudes de los usuarios utilizando Hadoop para procesamientos masivos y escalables

Presentada por

Mervyn Xavier Macías Martrus

Freddy Fernando de La Rosa

Octubre 2009

AGENDA

- Sistema de recomendaciones
- Algoritmos de recomendaciones
- Dataset
- Diseño
- Esquema map-reduce
- Implementación
- Alternativas de Ejecución
- Resultados
- Recomendaciones
- Conclusiones

Sistemas de Recomendaciones

- Un sistema de recomendaciones es un tipo específico de filtro de información que sugiere a los usuarios ítems o productos concretos basándose en sus preferencias

Ejemplos de Sistemas de Recomendaciones

- iTunes
- Pandora
- Last.fm
- Facebook
- Snooth

Componentes

- Usuarios
- Perfiles
- Recomendadores
- Valoración

Mecanismos de Retroalimentación

- Explícita: Cuando se aplica la recomendación explícita, el sistema otorga al usuario la oportunidad de calificar, dentro de un rango predefinido, los ítems que ha utilizado
- Implícita: El sistema obtiene retroalimentación implícita capturando la interacción del usuario sin que él lo note.

Métodos de Recomendación

- Recomendación colaborativa
- Recomendación basada en contenido
- Recomendación basada en demografía
- Recomendación basada en utilidad
- Recomendación basada en conocimiento

Recomendación Basada en Contenido

- En este caso la recomendaciones son hechas exclusivamente en base a los ítems que el usuario ha elegido en el pasado.

Recomendación Colaborativa

- Cuando se aplica recomendación colaborativa, las recomendaciones son hechas exclusivamente en base a los usuarios con gustos similares.
- Hay sistemas que aplican más de un método de recomendación:
  - FAB
  - SELECT

Desventajas de la Recomendación Basada en Contenido

- Se hace un análisis automático de los ítems, considerando atributos predefinidos pero se deja de lado otros atributos relevantes.
- Las sugerencias son hechas en función del historial de ítems elegidos por el usuario, por lo tanto el usuario sólo puede recibir recomendaciones que concuerden con su perfil

Desventajas de la Recomendación Colaborativa

- Cuando un nuevo ítem se añade al sistema, como no ha sido evaluado por ningún usuario, no hay forma de recomendarlo.
- Si un usuario particular no se identifica con los gustos de ningún otro usuario del sistema, no es factible hallar vecinos cercanos, y por lo tanto hacer recomendaciones.
- Se requiere un mínimo de usuarios para elaborar las predicciones

Ventajas de Recomendación Colaborativa sobre Basada en Contenido

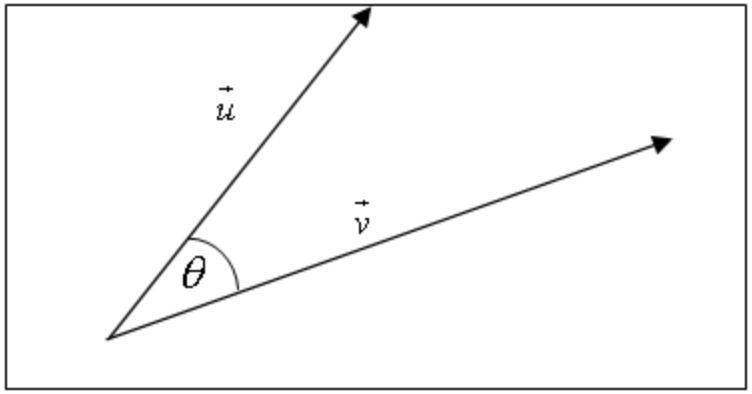
- En la recomendación colaborativa, los usuarios utilizan sus criterios para evaluar los ítems, cubriendo así más características de los ítems que evalúan.
- En recomendación colaborativa es posible que el usuario reciba recomendaciones que no se alineen a su perfil (pero sí a sus gustos).

Algoritmos de Recomendación Colaborativa

- Los algoritmos de recomendación colaborativa se basan en el producto punto entre dos vectores y en las fórmulas de correlación .

Producto Punto

$$\vec{u} \cdot \vec{v} = \left|\vec{u}\right|\left|\vec{v}\right| \cos\theta$$



Producto Punto

- Cuando θ es cero, los vectores apuntan en la misma dirección. Así, para valores de cercanos a cero, los vectores tienden a apuntar en la misma dirección.

Producto Punto

- Ejemplo:

	Ítems				
Usuario	**A**	**B**	**C**	**D**	**F**
1	4	5	0	0	0
2	5	6	0	5	0
3	5	0	4	0	8
4	0	0	5	0	0

Correlación

- El coeficiente de correlación de **Pearson** mide la relación lineal entre dos variables cuantitativas.

$$sim(u,v) = \frac{\sum_{i=1}^{m} \left(r_{u,i} - r_{\vec{u}}\right)\left(r_{v,i} - r_{\vec{v}}\right)}{\sigma_u \sigma_v}$$

Algoritmos de Recomendación Basado en Usuario

- Similitud

$$Sim\left(\vec{u_i}, \vec{u_j}\right) = \frac{u_i \cdot u_j}{\left|u_i\right|\left|u_j\right|}$$

- Recomendación

$$p_{a,i} = \frac{\sum_{k=1}^{n} r_{u_k,i} \cdot Sim\left(\vec{a}, \vec{u_k}\right)}{\sum_{k=1}^{n} Sim\left(\vec{a}, \vec{u_k}\right)}$$

Algoritmos de Recomendación Basado en Usuario

- Con la aplicación de la fórmula anterior para cada par de usuarios del sistema se genera una matriz de similitud. Por ejemplo, si la primera fila de la matriz de similitud contiene la siguiente información:

- u1: 0.5 u2 | 0.3 u3 | 0.2 u4 | 0.6 u7 | 0.9 u8

Algoritmos de Recomendación Basado en Ítems

- El principio es el mismo que en el caso de las recomendaciones basadas en usuario, la diferencia es que en este caso buscamos similitudes entre ítems en lugar de buscar similitudes entre usuarios.

Análisis de Algoritmos

- En general, para un dataset con n usuarios y m ítems, para cada usuario se deben realizar n-1 comparaciones, en total n(n-1). En el peor de los casos cada comparación implica m operaciones. Así, el tiempo de ejecución es del orden de mn^2.

Datasets

- Un dataset es una colección de datos presentados, por lo general, en forma tabular, de tal manera que cada columna representa una variable particular y cada fila corresponde a un miembro del dataset. La fila contiene los valores de las variables correspondientes a cada columna. Cada valor recibe el nombre de dato.

Ejemplos de Dataset

- MusicBrainz es un dataset de música mantenido por la comunidad de usuarios. Contiene datos como el nombre del artista, título del álbum lanzado y la lista de pistas que aparecen en un álbum lanzado.
- El dataset de Audioscrobbler contiene una columna de usuarios, una de artistas y el número de veces que cada usuario escuchó a cada artista.

Dataset de Audioscrobbler

	id_user text	id_artist text	value double precis
825	1000002	82	9
826	1000002	831	36
827	1000002	833	5
828	1000002	860	32
829	1000002	893	1
830	1000002	930	3
831	1000002	949	86
832	1000002	958	5
833	1000002	969	2
834	1000002	976	5
835	1000002	979	86
836	1000002	988	23
837	1000002	999	6
838	1000019	1000010	11
839	1000019	1000028	5
840	1000019	1000033	2
841	1000019	1000036	5
842	1000019	1000054	1

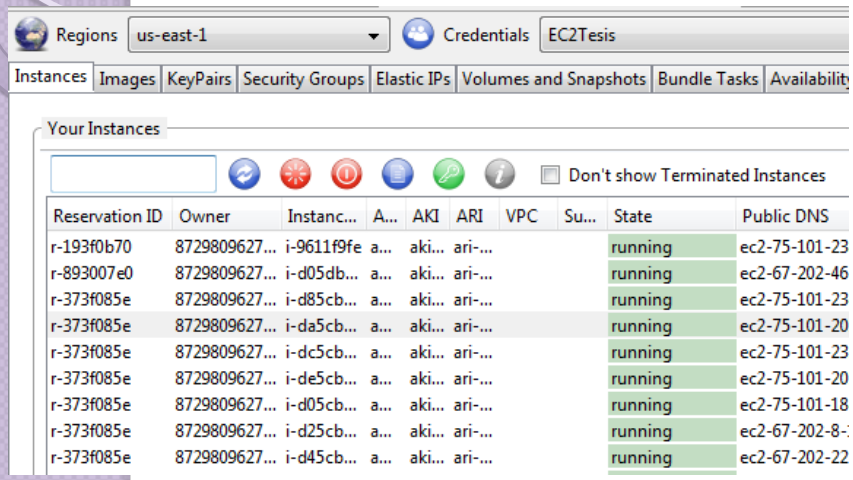
Herramientas

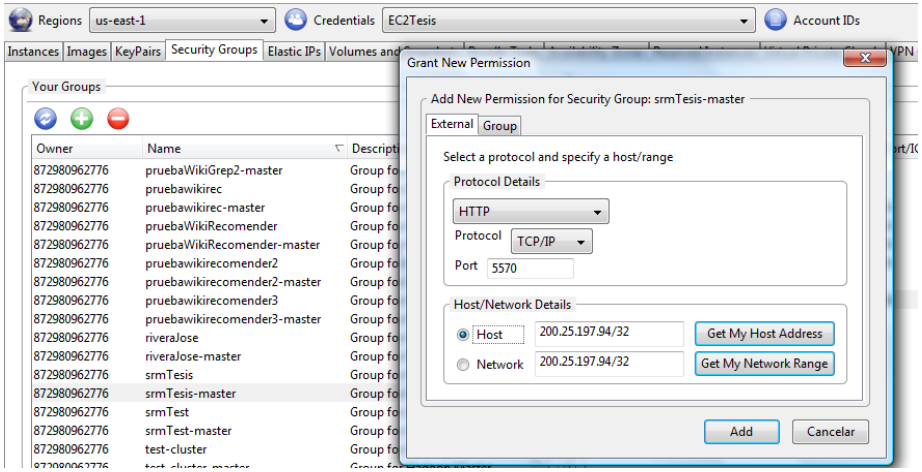
- Map-Reduce
- Hadoop
- Mahout
- Amazon Web Services - EC2
- Amazon Web Services - S3

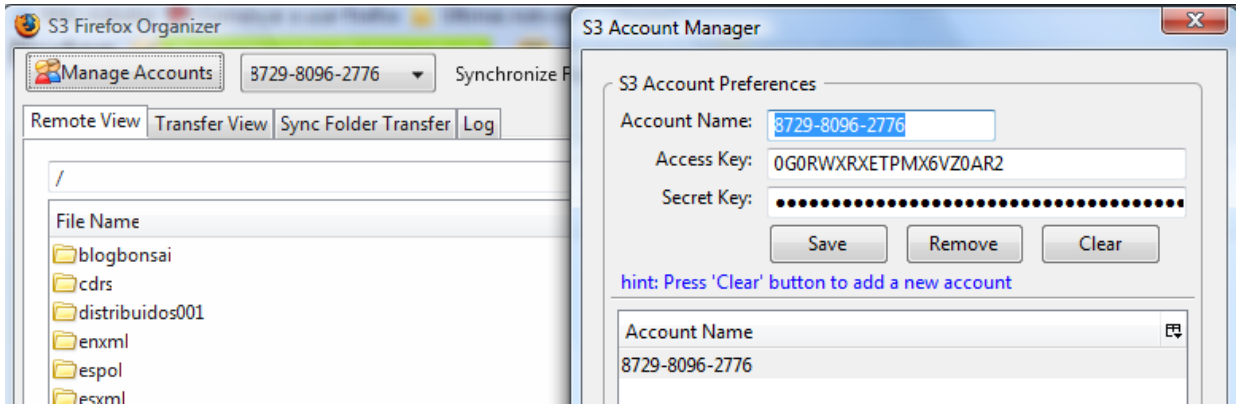
Diseño

- Requerimiento de hardware
- Requerimiento de software
  1. Putty
  2. Eclipse
  3. Firefox
     - Amazon S3 Organizer
     - Elasticfox
  4. Vmware
  5. Filezilla

Plug-in Firefox





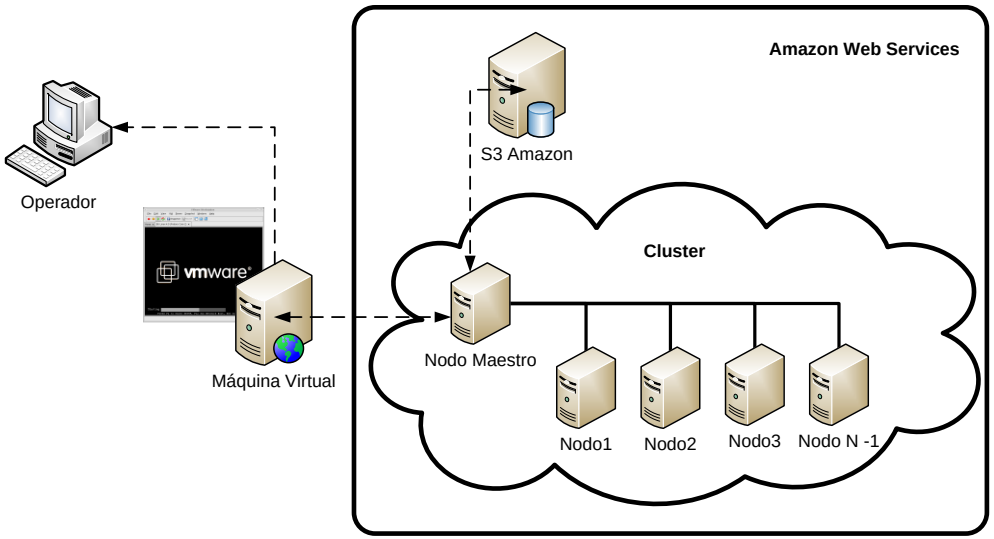


Vmware Workstation

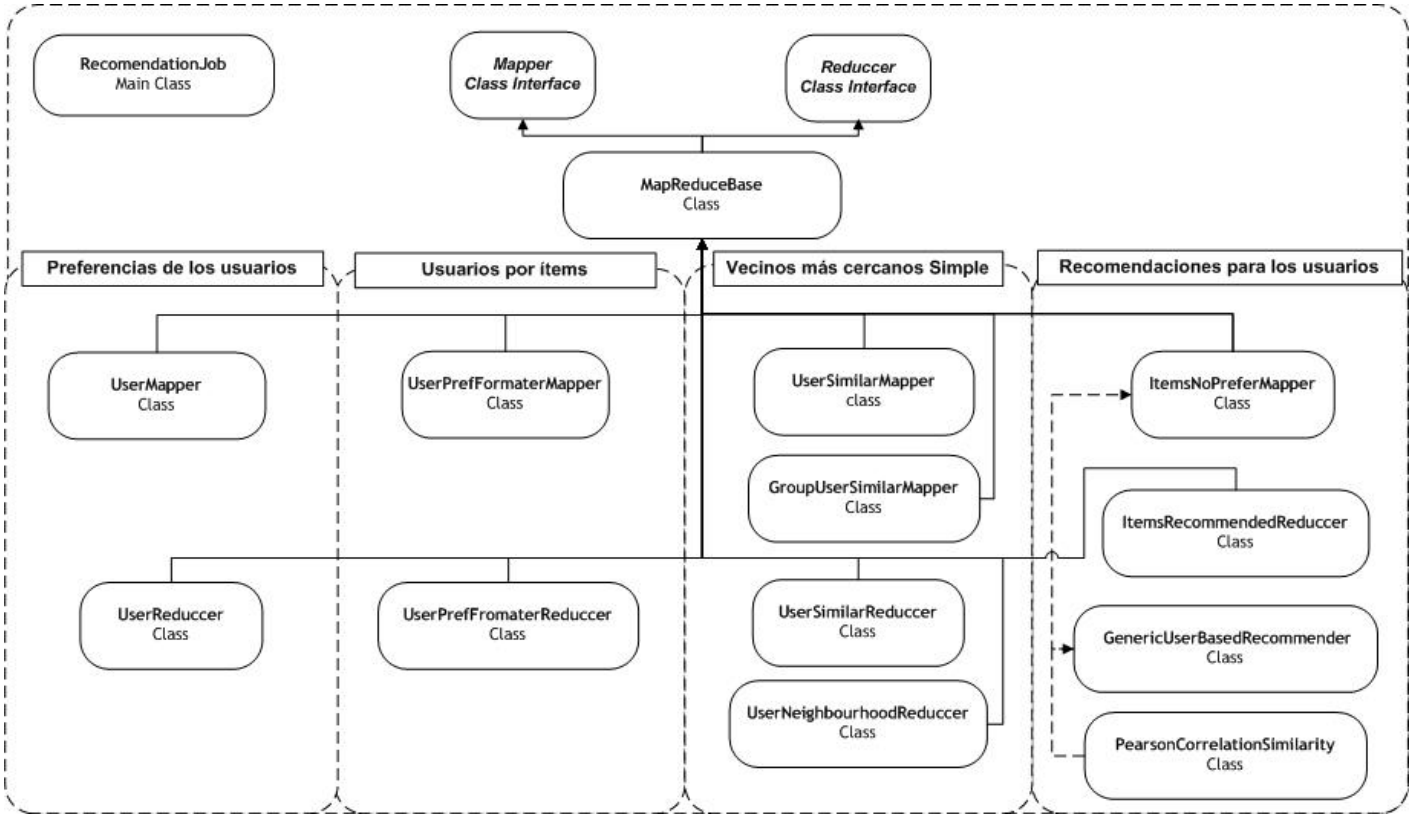
Name	Creator	Host OS(s)	License	Can boot an OS on another disk partition as guest	USB	GUI	Live memory allocation	3D acceleration	Snapshot of running system	Live migration
Virtual Server 2005 R2	Microsoft	Windows 2003, XP	Proprietary					No		
Windows Virtual PC	Microsoft	Windows 7	Proprietary	No	partially	Yes				
Virtual PC 2007	Microsoft	Windows Vista, XP	Proprietary	No	No	Yes	No	No		
VMware Server	VMware	Windows, Linux	Proprietary			Yes		No	Yes	Yes
VMware Workstation 6.0	VMware	Windows, Linux	Proprietary	Yes	Yes	Yes	Yes	Experimental support for DirectX 8	Yes	
VMware Player 2.0	VMware	Windows, Linux	Proprietary	No	Yes	Yes	Yes	Supported with VMGL		
Sun xVM VirtualBox	Sun Microsystems	Windows, Linux, Mac OS X (Intel), Solaris, eComStation	GPL version 2	Partial (since version 1.4, but unsupported)	Partial (for any host OS except Solaris)[8]	Yes	Yes	OpenGL 2.0	Yes (only on the closed-source edition)	Yes (only on the closed-source edition)

Infraestructura Tecnológica

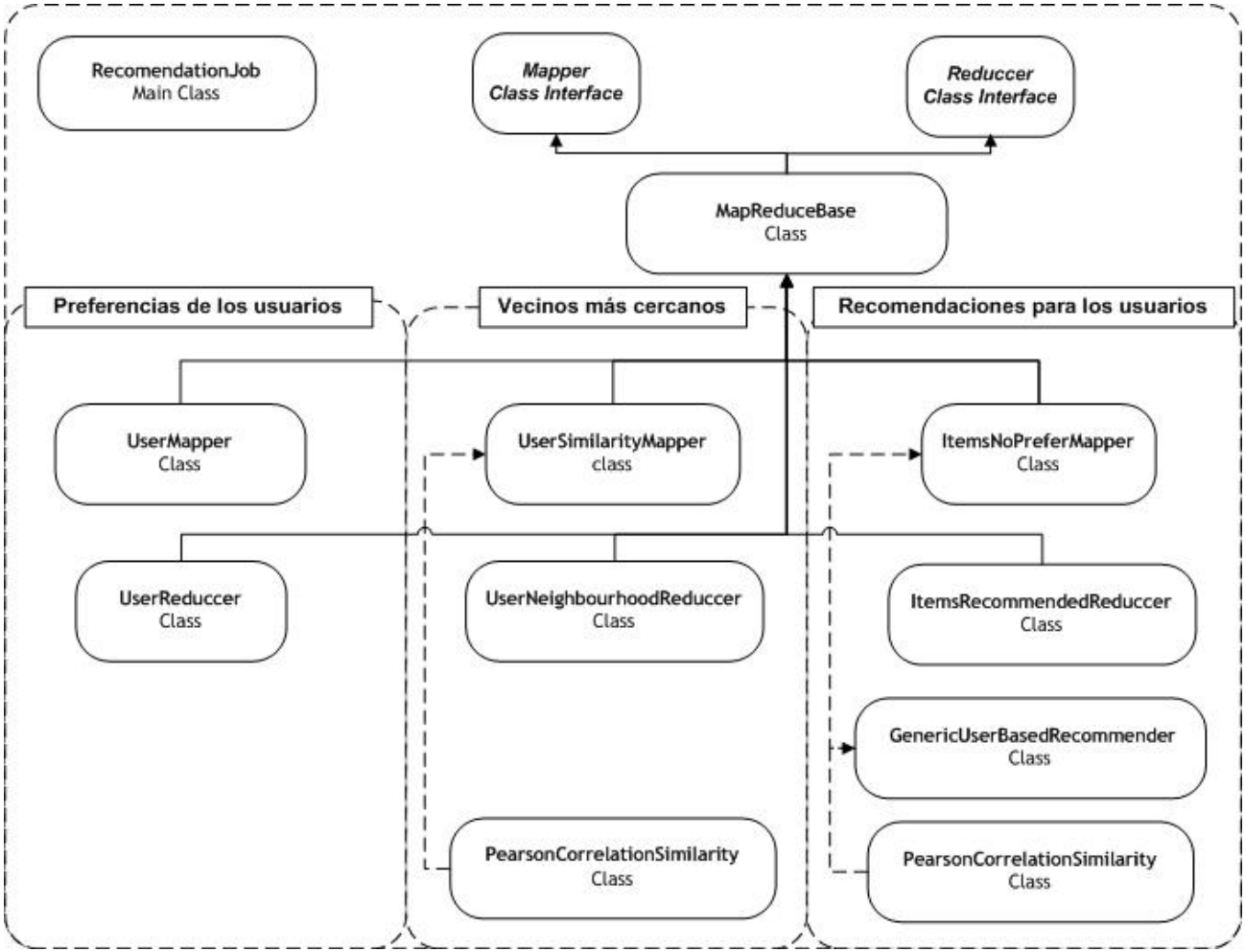
- Operador
- Máquina Virtual
- Nube EC2
- Nodo Maestro
- Nodos Esclavos
- Amazon S3



Producto Punto



Pearson

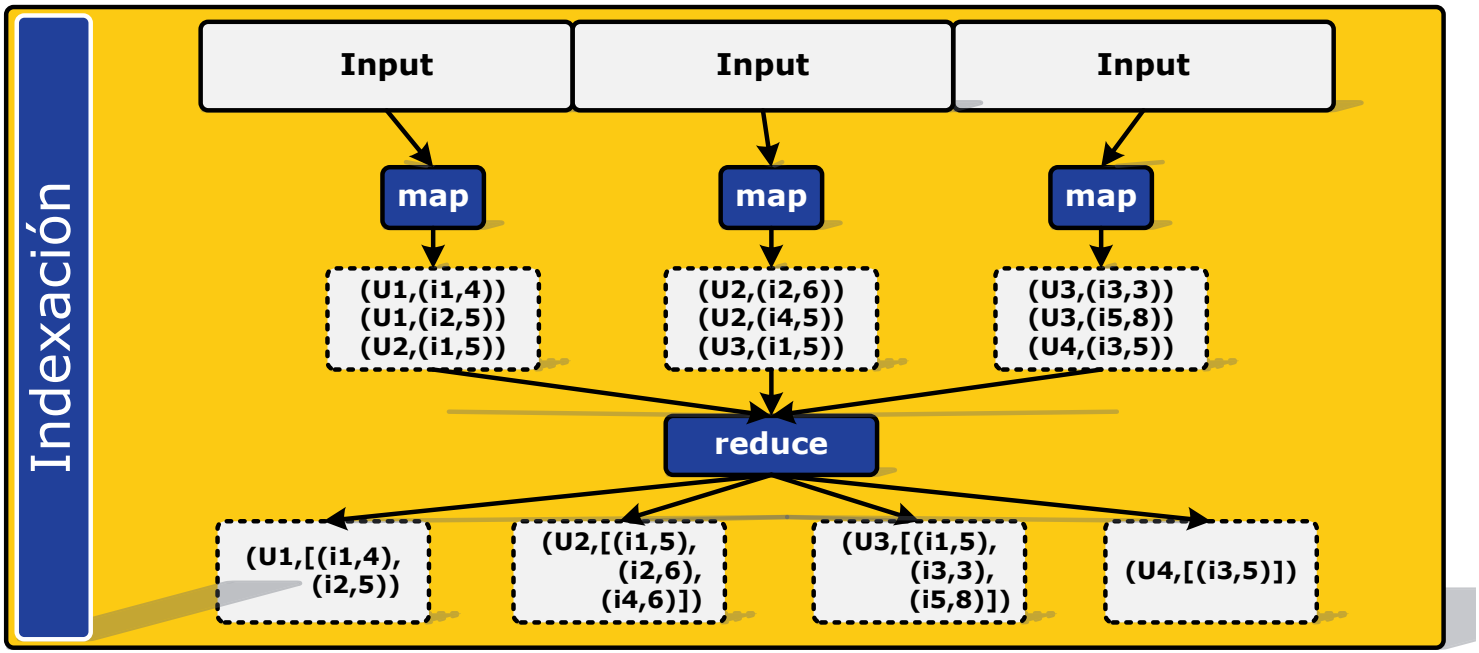

RecomendationJob
Main Class
Mapper
Class Interface
Reducer
Class Interface
MapReduceBase
Class
Preferencias de los usuarios
Vecinos más cercanos
Recomendaciones para los usuarios
UserMapper
Class
UserSimilarityMapper
class
ItemsNoPreferMapper
Class
UserReducer
Class
UserNeighbourhoodReducer
Class
ItemsRecommendedReducer
Class
GenericUserBasedRecommender
Class
PearsonCorrelationSimilarity
Class
PearsonCorrelationSimilarity
Class

Esquema map-reduce

- MapReduce es un modelo de programación para el procesamiento de grandes cantidades de información
- Función **"map"** procesa pares clave/valor para generar un conjunto de pares intermedios clave/valor, y una función **"reduce"** que agrupa todos los valores intermedios asociados con la misma clave

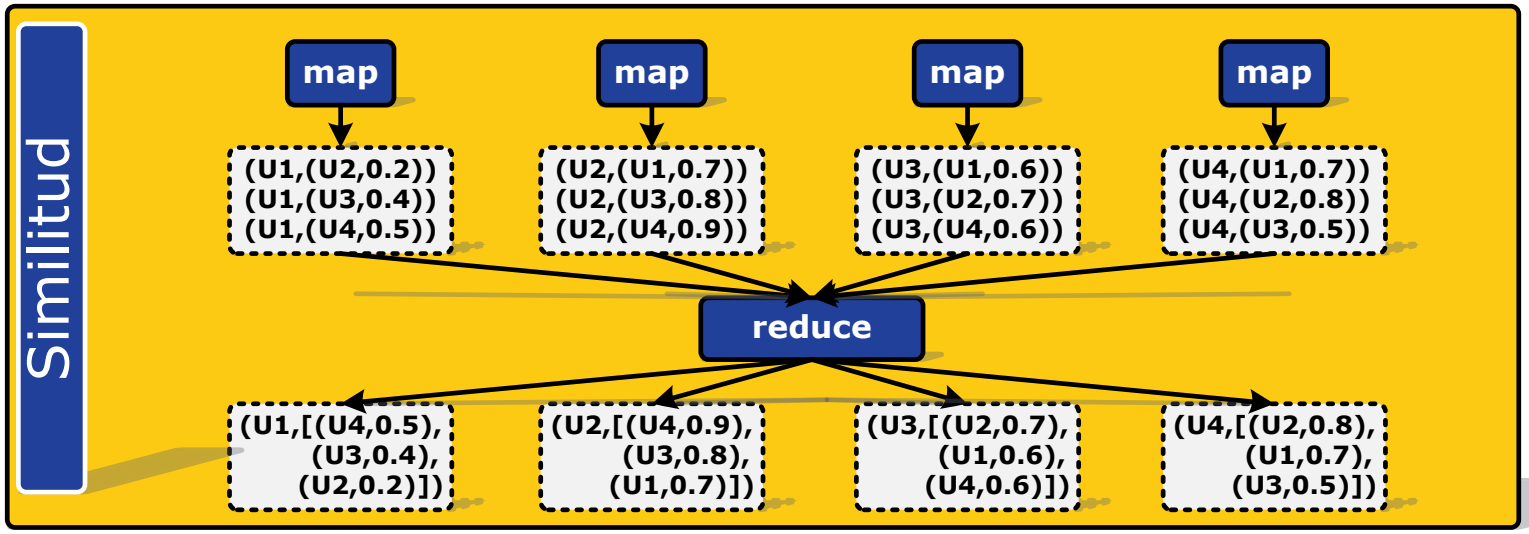
Esquema map-reduce: Indexación

- Pre-procesamiento de datos



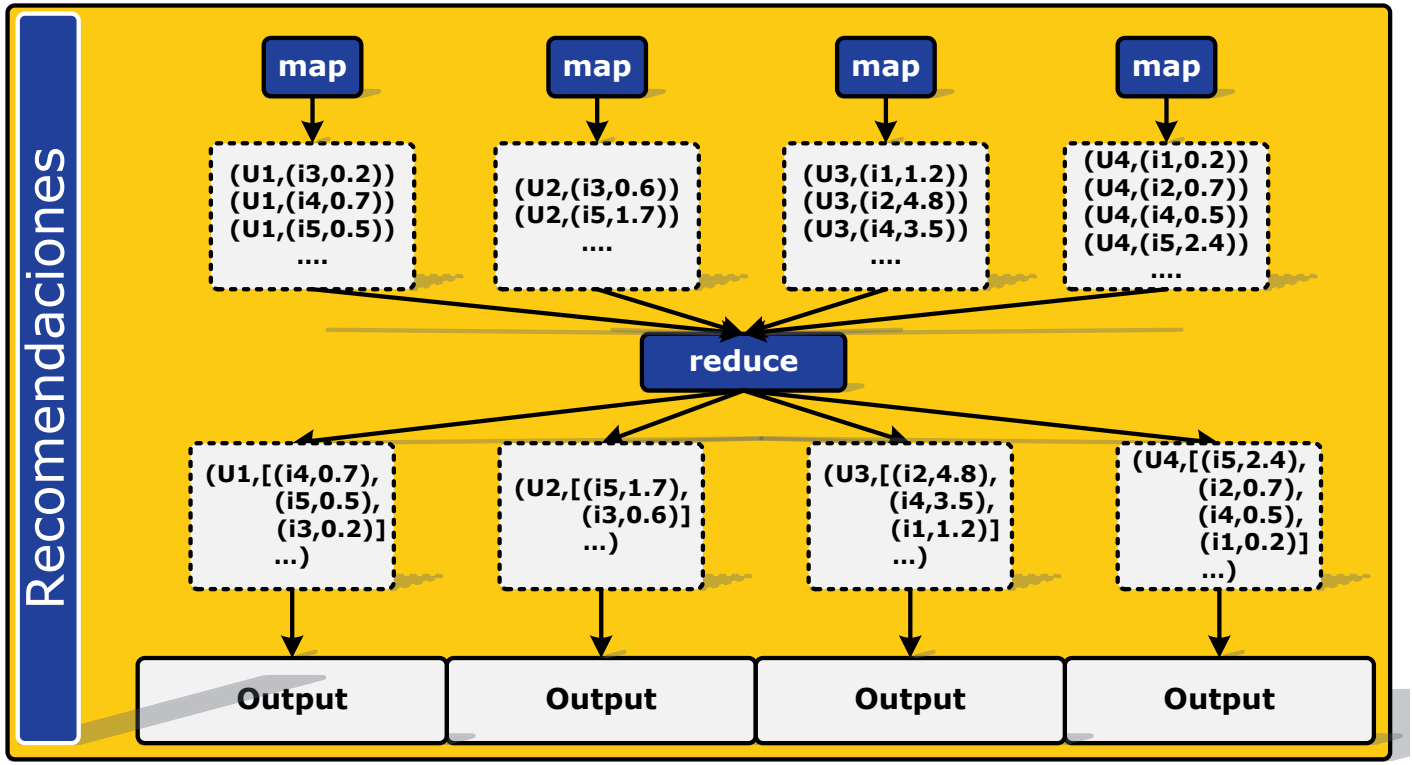
Esquema map-reduce: Similitud

- Elección de los vecinos más cercanos

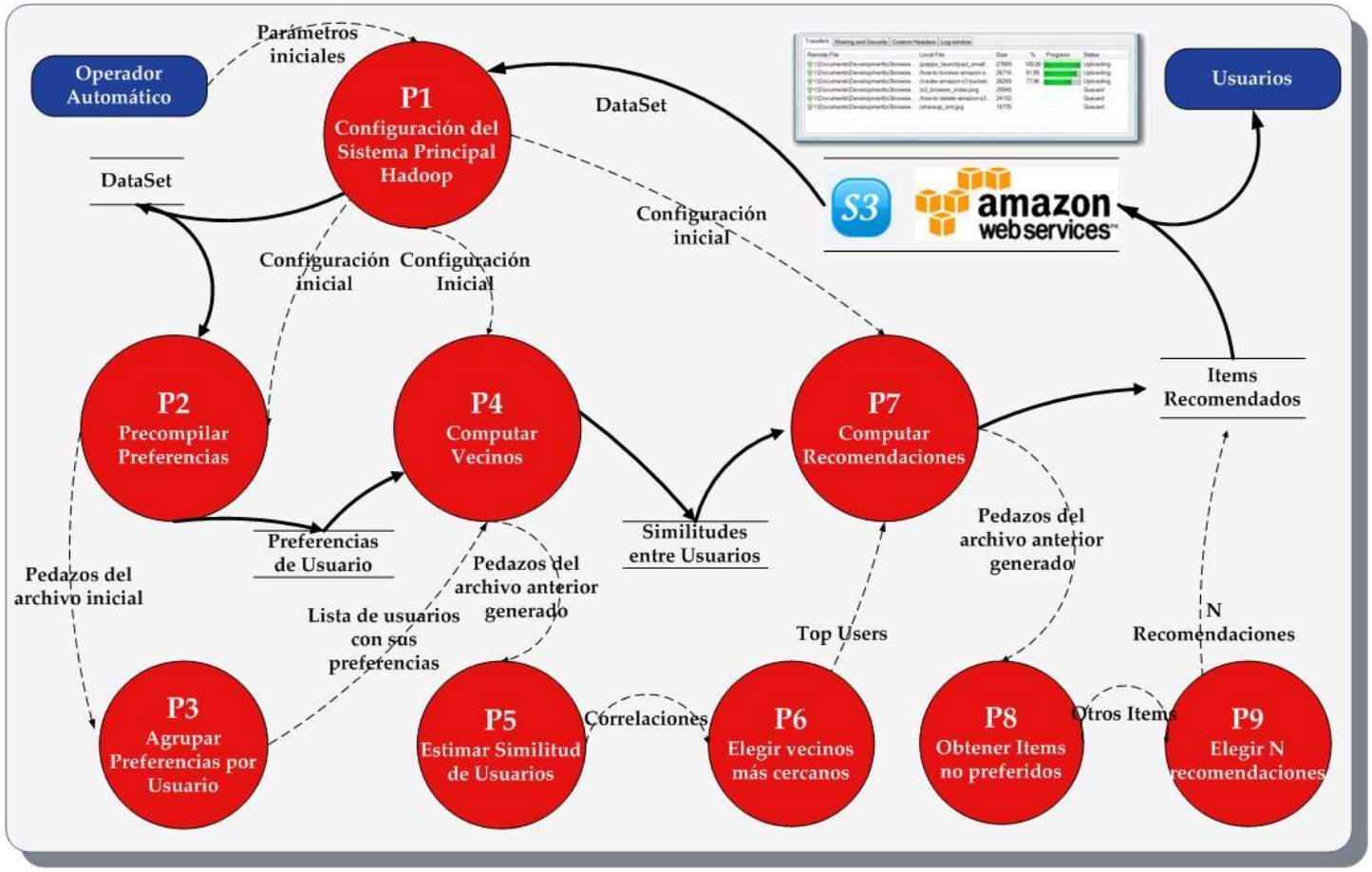


Esquema map-reduce: Recomendaciones

- Selección de ítems nunca antes visto.



Flujo de datos

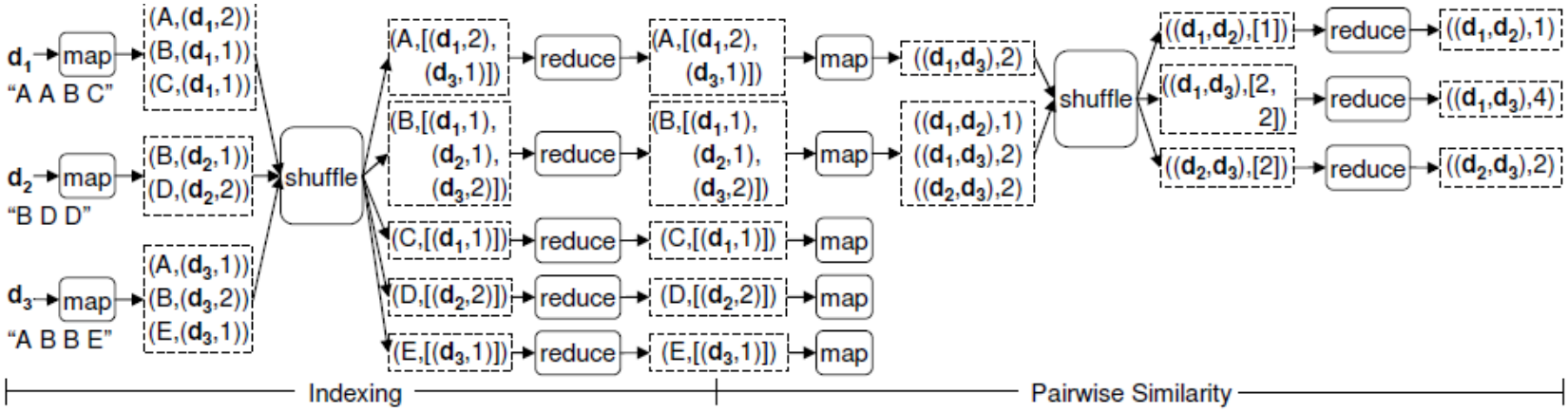


Recomendaciones usando Pearson

- **Proceso 1:** Configuración del Sistema Principal Hadoop
- **Proceso 2:** Pre-compilar Preferencias
- **Proceso 3:** Agrupar Preferencias por Usuario
- **Proceso 4:** Computar Vecinos
- **Proceso 5:** Estimar Similitud de Usuarios
- **Proceso 6:** Elegir vecinos más cercanos
- **Proceso 7:** Computar Recomendaciones
- **Proceso 8:** Obtener ítems no preferidos
- **Proceso 9:** Elegir N recomendaciones

Recomendaciones usando producto punto

- **Proceso 4:** Agrupar usuarios por ítem
- **Proceso 5:** Agrupar por tuplas de usuarios



Computo de los valores de similitud simple presentado por Tamer Elsayed,_Jimmy Lin,† and Douglas W. Oard

Configuraciones

- Parámetros de inicio:

  <input> Es la ruta de origen HFDS donde se almacena el dataset.

  <output> Es la ruta de destino HDFS donde se almacenan los resultados.

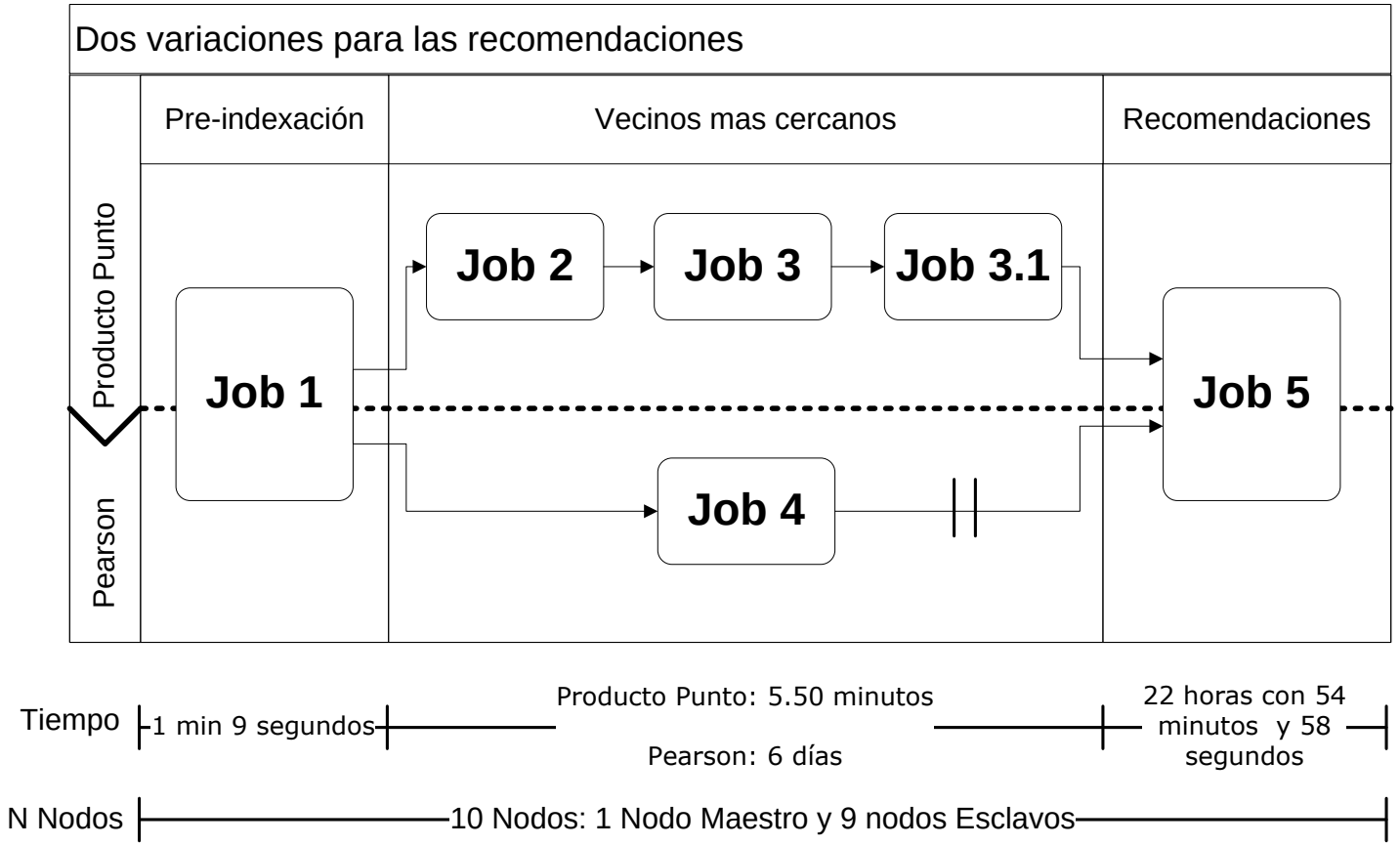
  <nvecindad> Es el número de vecinos más cercanos.

  <nrecomendaciones> Es el número de recomendaciones por usuario.

Configuraciones

- Variables de inicio clave/valor para el **map** o **reduce** (opcional). Ejemplo:
  - `jobConf.set(UserNeighbourhoodReduccer.NEAREST_NUSER_NEIGHBORHOOD, nvecindad);`
  - `jobConf.set(ItemsRecommendedReduccer.RECOMMENDATIONS_PER_USER, nrecomendaciones);`
- Carga de archivos en la cache distribuida (opcional). Ejemplo:
  - `DistributedCache.addCacheFile(new URI(inputPathDis), jobConf);`

Ejecución de Jobs con 2 alternativas

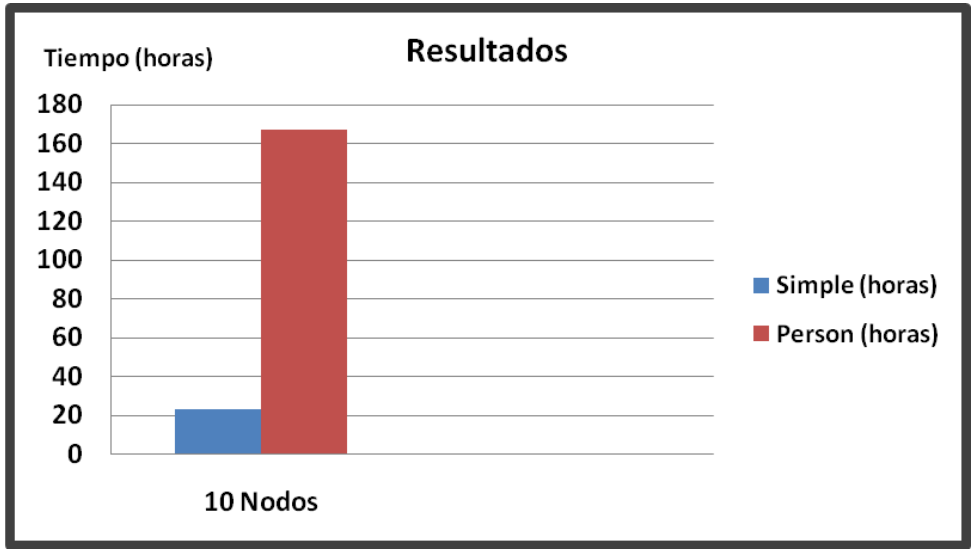


Tiempos de respuesta

Job	map tasks	reduce tasks	Tiempo map-reduce (min)
2	4	10	1.25
3	10	10	3.4
3.1	10	10	0.85
		Total	5.50

Proceso de obtención de vecinos más cercanos basado en una correlación simple.

Reducción de un 99.93% con respecto a la alternativa de la correlación de Pearson.



Costos

Descripción	Simple	Con Pearson
Constante (nodos)	$0.20	$0.20
Número de nodos en cluster	10	10
Duración cluster levantado (Horas)	24	169
Subtotal Nodos	$48.00	$338.00
Constante (Upload por GB)	$0.10	$0.10
Tamaño Upload (GB)	0.00000	0.00000
Subtotal Upload	$0.00	$0.00
Constante (Download por GB)	$0.17	$0.17
Tamaño Download (GB)	1	1
Subtotal Download	$0.17	$0.17
Costo Total Sesion	$48.17	$338.17

La correlación de Pearson de la librería de Mahout tiene un costo mucho mayor que utilizando la correlación simple .

Costo total del proyecto

Categoría	Valor
Saldo Inicial	$200.00
Costo Total EC2 (US)	$125.73
Costo Total Storage (US)	$0.81
Costo Total Proyecto	$126.54
Saldo Disponible	$73.46

Conclusiones

1. Los tiempos de ejecución de los algoritmos implementados aumentan con el cuadrado de la cantidad de usuarios y por ello presentan problemas de escalabilidad.
2. El costo del algoritmo de Pearson es tan alto en términos monetarios, que supera 8 a 1 al algoritmo basado en producto punto.
3. Aplicando producto punto no se obtienen recomendaciones para usuarios que han calificado ítems que otros usuarios no han calificado, Pearson resuelve este problema aplicando inferencias.

Recomendaciones

1. La poca cantidad de nodos compartidos es una limitante. El producto punto se constituye en una alternativa viable para el procesamiento de las similitudes.
2. Para el seguimiento futuro de estos casos se puede buscar una alternativa al método de producto punto en el que se apliquen inferencias y utilice el esquema map-reduce para escalabilidad.