Sobre la Factibilidad de Usar Técnicas de Minería de Flujos de Trabajo Para Inferir Grafos de Ejecución en Sistemas de Procesamiento Distribuido

Gladys E. Carrillo, Cristina L. Abad

Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral, Guayaquil–Ecuador
gecarri@espol.edu.ec, cabad@fiec.espol.edu.ec

Resumen. En el presente trabajo, se considera el problema de la evaluación de nuevas mejoras a plataformas de procesamiento distribuido. Específicamente, se considera la evaluación utilizando cargas de trabajos reales publicadas por empresas con grandes clústeres de datos. Estas evaluaciones son comúnmente utilizadas por investigadores ya que permiten demostrar la utilidad de sus proyectos de investigación. Sin embargo, presentan el problema que hasta el momento no se ha liberado ninguna traza de trabajos que contenga información de las dependencias existentes entre los mismos.

La metodología propuesta está basada en técnicas de minería de flujos de trabajo (worfklow mining) para obtener cargas de trabajos distribuidos con dependencias realistas entre trabajos. Esta metodología permite obtener esta información a través del minado automatizado de trazas que carecen de información de dependencias entre trabajos. Finalmente, se demuestra que la metodología propuesta es capaz de encontrar flujos de trabajo realistas en trazas publicadas por Google.

Palabras clave: Procesamiento distribuido, clústeres, minería de datos, Hadoop, flujos de trabajo, cargas de trabajo.

1 Introducción

Los grandes volúmenes de datos que se generan a la par del crecimiento exponencial de la Internet, requieren igualmente sistemas de procesamiento con altas capacidades que permitan obtener beneficio de dicha información. Esto, conjugado con la necesidad de mantener elevados niveles de disponibilidad y tolerancia a fallos, ha derivado en el desarrollo de sistemas de procesamiento distribuido, que es un área de investigación y desarrollo muy activa.

Existen varios sistemas de procesamiento distribuidos implementados en desarrollo tales como MapReduce [10], Hadoop [7], Dryad [13], Spark [24], entre otros. Además, sobre los sistemas más populares continuamente se proponen mejoras [15, 19, 23, 3, 5, 6, 12] que requieren ser verificadas utilizando cargas de trabajo reales que determinen si las modificaciones propuestas realmente producen incrementos en el rendimiento en un entorno determinado.

La evaluación de sistemas de procesamiento distribuido también es importante al momento de decidir cual se debe utilizar para una determinada aplicación, puesto que, dependiendo de las características de la carga de trabajo de los casos particulares, puede variar el rendimiento de cada uno de los sistemas disponibles.

Lastimosamente, las pruebas de rendimiento(benchmarks)existentes no reproducen flujos de trabajo (workflows) realistas, debido a que ninguna empresa ha liberado públicamente trazas de ejecución de flujos de trabajo en plataformas distribuidas. Esto lleva a los investigadores a recurrir a flujos de trabajo artificiales, los cuales podrían ejecutar las cargas de trabajo en un orden que asume cero dependencias, o a recurrir a "inventarse" flujos de trabajo que probablemente no reflejen situaciones realistas.

El no tomar en cuenta las posibles relaciones de precedencia que pueden existir entre los distintos trabajos modifica el rendimiento del sistema, puesto que la ejecución paralela de dos o más trabajos toma menos tiempo que el requerido bajo una dependencia, en donde los mismos trabajos serian ejecutados secuencialmente. Es decir, el realizar pruebas usando cargas de trabajo con dependencias, pero ignorando las mismas, genera ejecuciones no validas ya que violarían las dependencias de la carga de trabajo original.

La Figura 1(b) ilustra el problema de violación de dependencias que puede ocurrir cuando se utiliza trazas de ejecución de trabajos sin información de dependencias. Como se puede observar, si la traza no contiene dicha información, se puede violar la dependencia $C \rightarrow D$. Nótese que una posible razón por la que C no haya sido ejecutada en paralelo con B seria si C necesita de recursos no disponibles al momento en que C desea ejecutarse (por ejemplo, están en uso por B). Esto conllevaría a que C se ejecute más tarde que en la traza original. Para respetar la dependencia, debería postergarse la ejecución de D hasta que C termine. Pero dado que no se puede saber que existe una dependencia entre C y D, entonces se puede ejecutar D antes de tiempo, violando la dependencia entre ambos trabajos. En la Figura 1(c) se muestra un ejemplo de una situación ideal en la que la traza *si* contiene la información de las dependencias entre trabajos, permitiendo así ejecuciones que no violen dichas dependencias.

En el presente estudio, se explora el uso de técnicas de minería para la extracción de flujos de trabajo a partir de trazas reales de ejecución de sistemas de procesamiento distribuido, que puedan ser luego incluidos en la ejecución de pruebas de rendimiento de este tipo de sistemas. La minería de flujos de trabajo consiste en revisar registros de ejecución de actividades para extraer información que permita reflejar relaciones de dependencia u orden de ejecución entre dichas actividades.

2 Técnicas de Minería de Flujos de Trabajos

Los workflows o flujos de trabajo son una secuencia ordenada de actividades o pasos para llevar a cabo una tarea determinada. Tienen aplicación en distintos campos como ingeniería, administración de empresas o informática. Pero quizás donde más se han utilizado es en el ámbito empresarial, puesto que sirven para la definición o modelamiento de los procesos del negocio. El uso de flujos de trabajo puede mejorar

el control de la ejecución de los procesos y el flujo de información en las relaciones con clientes y proveedores.

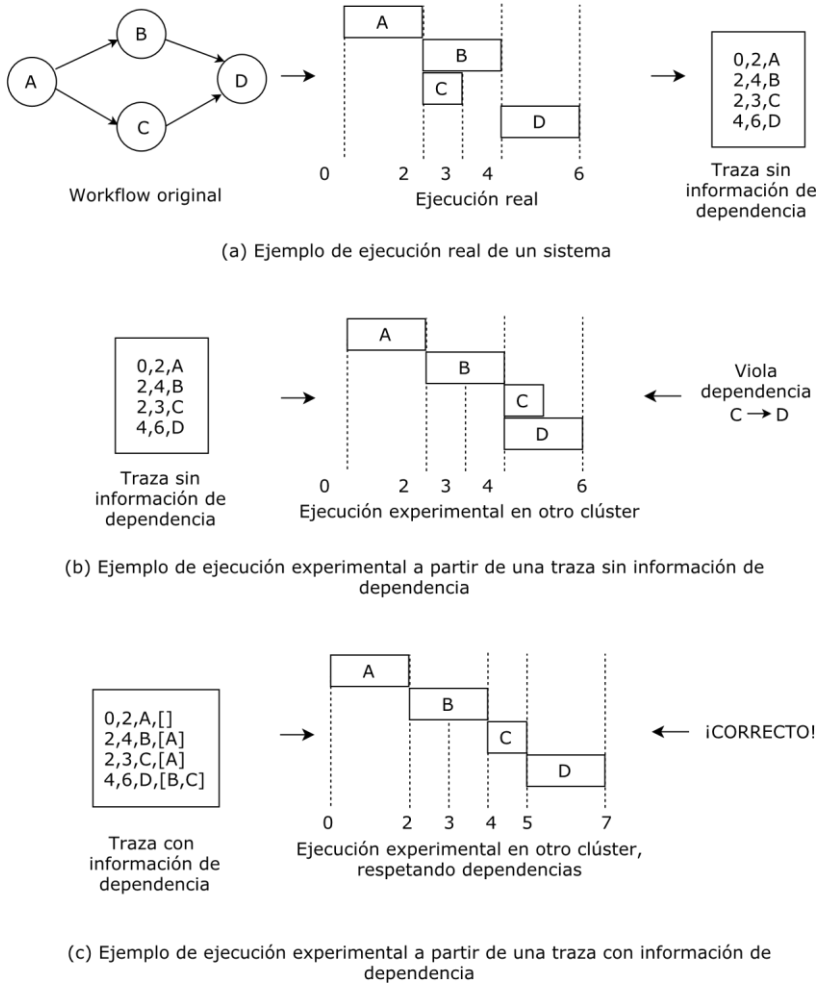

Figura 1: La secuencia de imágenes muestra la diferencia de ejecutar una traza sin y con relación de dependencia. En (a) la ejecución real genera una traza sin información de dependencia. En (b) la traza sin información de dependencia se usa para una ejecución experimental que viola la dependencia $C \rightarrow D$. En (c) se realiza una ejecución experimental a partir de una traza con información de dependencia.

Muchas veces, la creación de los flujos de trabajo no es una tarea trivial y, para su correcta ejecución, requiere de la participación de un experto en la rama. Por esta razón, el desarrollo de técnicas para su generación autónoma es un campo activo de

investigación. Una de estas técnicas es la minería de flujos de trabajo o minería de procesos, que consiste en revisar registros de actividades de sistemas para extraer información que permita reflejar el orden en que se ejecutan las actividades (tareas, procesos, etc.) y la relación entre ellas [4].

Las técnicas de minería de flujos de trabajo se pueden aplicar sobre los registros de actividades del sistema para lograr diferentes propósitos. En un primer caso, cuando la aplicación o el sistema no tiene definido explícitamente sus componentes y su funcionamiento, el resultado obtenido luego de aplicar las técnicas de minería ayuda a descubrir la estructura del funcionamiento de los procesos para entender el comportamiento del sistema. Por otro lado, cuando en el sistema ya están definidos los flujos de trabajo, la minería se utiliza como apoyo para verificar si lo que se diseño es en realidad lo que se está ejecutando y enfocarse en corregir aquellos puntos que pueden afectar en gran medida el rendimiento general del sistema [1].

Algunos enfoques se han propuesto para la minería de flujos de trabajos. El algoritmo alpha [2] construye una red de Petri[1] analizando las relaciones causales que pueden existir entre las actividades y ha sido utilizado por otros algoritmos [21] y aplicaciones [11] como un punto de partida en el proceso de descubrimiento de flujos de trabajos. Otro enfoque considera extraer flujos de trabajos de registros de eventos intercalados [14], contradiciendo la presunción común de que las actividades se encuentran totalmente ordenadas en los registros; esta técnica identifica dependencias temporales entre pares de eventos, calculando valores de confianza a través de la probabilidad de ocurrencia de un evento con respecto al otro. Con estas dependencias se construye un flujo de trabajo básico que luego es refinado para obtener el flujo de trabajo final que contendrá relaciones más estructuradas como bifurcación/unión, lazos y transiciones directas.

Un enfoque estadístico de minería de procesos “Heuristics Minera” es propuesto por Weijters y van der Aalst [20]. Esta técnica examina la totalidad del archivo de registros descubriendo dependencias causales entre actividades a través de la aplicación de fo´rmulas que consideran el número de veces que una actividad precede a otra. Una adaptación de este algoritmo es propuesto por Burattin y Sperduti [8] para considerar los intervalos de tiempo de cada actividad al momento de realizar el proceso de minería. A diferencia del algoritmo “Heuristics Minera”, que considera para el ordenamiento de las actividades solo el tiempo de inicio o de fin de cada actividad, el algoritmo “Heuristics miner” propuesto en [8], plantea usar la duración de cada actividad cuando esta información esta´ disponible. Considerar los intervalos de tiempo, permite descartar dependencias causales entre dos actividades al momento que se superponen en el tiempo.

Zeng et al. [25] proponen el uso de técnicas de minería de datos para descubrir flujos de trabajo trans-organizacionales. Las técnicas de minería se aplican en primer lugar a los registros del sistema de cada organización. Cada log es examinado para calcular la duración de las actividades y descubrir las relaciones temporales y estructurales entre ellas. Con estas relaciones construyen una red de Petri extendida que

[1] Una red Petri es una gra´fica que ayuda a modelar la estructura de un sistema.

la denominan RM WF Net, que permite representar el intercambio de mensajes y la asignación de recursos entre los flujos de trabajo. Para la integración de los flujos de trabajo descubiertos en cada organización definen cuatro patrones de coordinación. Estos patrones se determinan combinando los registros del sistema entre pares de organizaciones para descubrir la coordinación de actividades sincronizadas, intercambio de mensajes, recursos compartidos y procedimientos. El resultado de esta integración es un flujo de trabajo general que refleja los procesos en todas las organizaciones asociadas.

3 Solución

La propuesta es el uso de técnicas de minería para la extracción de flujos de trabajo a partir de trazas reales de ejecución de sistemas de procesamiento distribuido, que puedan ser luego incluidos en la ejecución de pruebas de rendimiento de este tipo de sistemas. Como prueba de concepto, se seleccionó la técnica propuesta por Lou et al. [14], debido a que sus autores han publicado una herramienta de software "State-Machine Simulator and Minera" para su fácil aplicación. En un futuro, se puede comparar esta técnica con otras relacionadas para poder determinar la que resulte más adecuada al problema estudiado en este trabajo.

El programa de minería recibe como parámetros dos archivos de entrada. El primer archivo contiene los registros del sistema, donde cada línea del archivo representa una secuencia de tareas o eventos como se muestra en la Figura 2. El segundo archivo es un mapeo del orden en que estas actividades se presentan en los registros del sistema (ver Figura 3).

```
A1,A2,A3,A4,A6,A7,A8,A9,A10
A1,A4,A5,A4,A6,A7,A8,A9,A10
A1,A2,A3,A5,A6,A7,A5,A6,A10
```

Figura 2: Ejemplo del archivo de registros del sistema de entrada para la aplicación "StateMachine Simulator and Minera".

```
A1 0
A2 1
A3 2
A4 3
A5 4
```

Figura 3: Ejemplo del archivo de mapeo para la aplicación "State-Machine Simulator and Minera".

Luego de realizar el procesamiento con los parámetros de entrada, el resultado se escribe en un archivo, indicando el número de estados y transiciones, el estado inicial y final y el detalle de todas las relaciones. La Figura 4 muestra un ejemplo del archivo de salida.

La hipótesis que se plantea es que esta técnica de minería puede ser utilizada para encontrar flujos de trabajos realistas en las trazas de sistemas de procesamiento dis-

```
--------------------------------
---8 states
14 transitions
--------------------------------
--S0 ROOT
S1 COMMON
S2 COMMON
...

S7 END
--------------------------------
---
S0 A1 S1
S2 A3 S3
S7 A2 S2
```

Figura 4: Ejemplo del archivo de salida de la aplicación “State-Machine Simulator and Minera”.

tribuido que carecen de esta información. Con este objetivo, se proponen los siguientes pasos:

1. Recolectar las trazas que se encuentren disponibles del sistema de procesamiento distribuido.
2. Examinar y pre-procesar las trazas para obtener los archivos de entrada requeridos por el software de minería de flujos de trabajo.
3. Minar los archivos pre-procesados utilizando alguna herramienta de minería de flujos de trabajo.
4. Analizar el archivo de salida para descubrir si los estados y transiciones reflejan algún flujo de trabajo.

En la siguiente Sección se implementan estos pasos para una traza y una herramienta específica, y se evalúa la factibilidad de la solución propuesta.

4 Evaluación

Para la evaluación de la propuesta se utilizaron las trazas de la carga de trabajo de Google [22, 17, 18, 16] que representa información de una celda de computo de un clúster de 12500ma´quinasen un periodo de 29 días. Una celda es un grupo de máquinas, por lo general alojadas en un solo clúster, que comparten un mismo sistema de administración encargado de asignar las actividades entre ellas [17]. Estas actividades se asignan a las celdas como trabajos (jobs), donde cada trabajo puede constar de una o más tareas.

Para la evaluación se utilizó´ como fuente de datos la tabla de eventos de trabajos (jobs), que contiene la siguiente información: marca de tiempo, id del trabajo, tipo de evento, nombre de usuario, clase de planificador, nombre del trabajo, nombre lógico del trabajo.

Para generar un archivo compatible con el formato de entrada del programa indicado en la Sección anterior, se realizó el siguiente pre-procesamiento de los datos:

1. Se seleccionan los trabajos que tienen en el campo tipo de evento, el valor de que corresponde al estado de terminado (finish).
2. Por cada trabajo terminado se va contando el número de veces que el trabajo está presente en el archivo.
3. Luego de examinar la totalidad del archivo, se seleccionan aquellos trabajos que estuvieron presentes más de una vez. No se puede inferir dependencias de trabajos que solamente están presentes una vez en la traza ya que no habría información que lo soporte (confirme).
4. Se agrupan los trabajos con la misma cantidad de ejecuciones. Esta heurística permite descartar trabajos que lógicamente no podrían depender entre sí[2].
5. Con la combinación de todos los pares de trabajos que aparecen la misma cantidad de veces, se realiza un nuevo archivo, donde cada línea tendrá como inicio y fin el par de trabajos y entre ellos los trabajos que aparecen entre la ejecución de ambos.

Las Figuras 5 y 6 muestran dos ejemplos de cómo lucen los datos una vez que han sido pre-procesados. Para facilitar la lectura, los nombres complejos de los trabajos fueron representados de la forma J[número], donde [número] es un número secuencial que identifica a los trabajos.

[2] Una limitacio´n de este paso es que no considera que hay flujos de trabajo que puede que no hayan sido registrados desde su inicio a fin, ya sea al inicio o al final de la traza.

```
J1,J3,J4,J5,J6,J7,J8,J2
J1,J3,J2
J1,J3,J10,J11,J12,J13,J14,J6,J15,J7,J16,J17,...,J76,J7
7,J2
J1,J3,J77,J48,J78,J2
J1,J3,J2
J1,J3,J2
J1,J3,J77,J2
J1,J3,J79,J80,J81,J82,J83,J84,J2
J1,J3,J85,J86,J87,J88,J89,J57,J24,J2
```

Figura 5: Archivo generado para dos trabajos con nueve ocurrencias en el archivo de registros.

```
J1,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,J14,J15,...,J3
4,J2
J1,J3,J35,J12,J2
J1,J3,J36,J37,J8,J2
J1,J3,J38,J39,J33,J40,J41,J42,J2
J1,J3,J28,J23,J43,J44,J45,J46,J26,J29,J27,J33,J47,J2
```

Figura 6: Archivo generado para dos trabajos con cinco ocurrencias en el archivo de registros.

La salida de la ejecución del programa para los archivos de las Figuras 5 y 6 se muestra en las Figuras 7 y 8. Al utilizar herramientas gra´ficas para representar estas salidas a manera de grafos, fue posible descubrir los flujos de trabajo que se muestran en las Figuras 9 y 10.

```
--------------------------------
--90 states
95 transitions
--------------------------------
--S0 ROOT
S1 COMMON
S2 COMMON
S4 COMMON
....

S99 END
--------------------------------
---
S0 J1 S1
S2 J4 S99
S4 J8 S99
S6 J85 S99
S8 J86 S99
S10 J87 S99
....
S77 J41 S78
S78 J42 S80
S80 J2 S82
```

Figura 7: Salida del programa al minar el archivo de la Figura 5.

En las gra´ficas se visualiza posibles relaciones de dependencias entre los trabajos. Cabe recalcar que, al no tener una manera para confirmar si los flujos de trabajo encontrados son reales, se los denomina “realistas”. Es decir, las informaciones disponibles en las trazas permiten deducir que los flujos de trabajo hallados son posibles workflows reales, mas no se puede confirmar que lo sean.

```
--------------------------------
--46 states
47 transitions
--------------------------------
--S0 ROOT
S1 COMMON
S2 COMMON
S4 COMMON
....

S99 END
--------------------------------
---
S0 J1 S1
S2 J4 S82
S4 J5 S82
S6 J6 S82
....
S76 J40 S77
S77 J41 S78
S78 J42 S80
S80 J2 S82
```

Figura 8: Salida del programa al minar el archivo de la Figura 6.

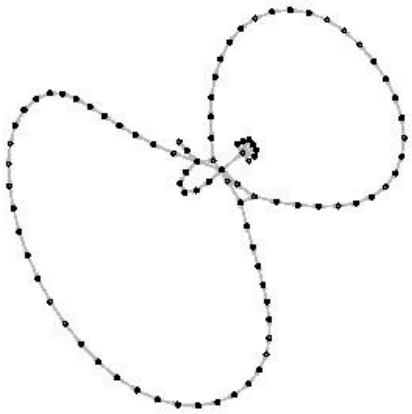

Figura 9: Flujo de trabajado generado a partir de la salida de la Figura 7.

Otra limitación es que posiblemente las trazas contienen otros flujos de trabajos reales que no fueron descubiertos ya que la documentación de las trazas indica que mu-

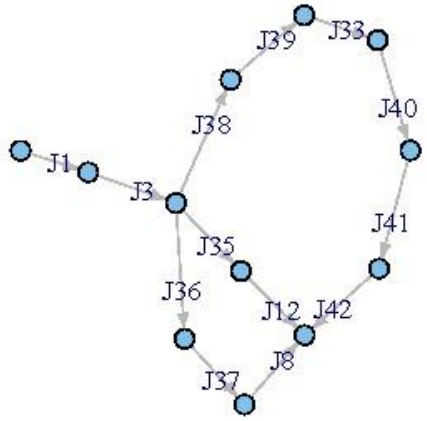

Figura 10: Flujo de trabajado generado a partir de la salida de la Figura 8.

chos de los trabajos tienen nombres únicos para cada ejecución del trabajo, impidiendo así que el software de minería distinga que dos ejecuciones pertenecen al mismo trabajo.

Con base en las dos limitaciones indicadas en el párrafo anterior, se recalca que lo ideal sería que cuando una empresa libere trazas de ejecución de trabajos (como lo hizo Google con su clúster data), incluya también con las mismas, cualquier información de dependencias entre los trabajos. Sin embargo, considerando que es imprescindible que se utilicen flujos de trabajo en las evaluaciones de planificadores y otras mejoras a las plataformas de procesamiento distribuido, se recomienda que hasta que esto ocurra, se utilice la metodología descrita en el presente trabajo para generar datasets con flujos de trabajo realistas que permitan realizar evaluaciones más efectivas de las mejoras propuestas a cualquier plataforma de procesamiento distribuido.

5 Recomendaciones

Con base en el estudio realizado en el presente trabajo, se emiten las siguientes recomendaciones para empresas, desarrolladores de plataformas de procesamiento distribuido e investigadores de sistemas distribuidos:

- Evaluadores de sistemas de procesamiento distribuido: Se recomienda el uso de flujos de trabajo realistas como parte de las entradas para las pruebas de rendimiento. La técnica descrita en este trabajo hace posible descubrir un flujo de trabajo realista.
- Empresas con grandes clústeres de procesamiento distribuido: Se recomienda liberar las cargas de trabajo junto con los flujos de trabajo para ayudar a los evaluadores a considerar situaciones más reales en las pruebas de rendimiento.

Si no les es posible liberar sus flujos de trabajo completos, se recomienda que en las trazas de las cargas de trabajo se preserve cierta información, como los identificadores o los nombres de los trabajos, los cuales pueden ayudar a inferir ciertos comportamientos de los datos.

- Desarrolladores de plataformas de sistemas distribuidos: Se recomienda incluir en sus mecanismos de generación de logs o bitácoras del sistema, registros que reflejen flujos de trabajos ejecutados y que faciliten la configuración de los mismos en la plataforma de procesamiento, de tal manera que luego resulte muy sencillo generar trazas que contengan esta información.

6 Trabajo futuro

Se va a comparar la ejecución de cargas de trabajo de sistemas distribuidos con un benchmark que no considera los flujos de trabajo como SWIM [9] para Hadoop y una versión modificada de SWIM que permite reproducir flujos de trabajo y demostrar que los resultados de rendimiento del Hadoop son diferentes, demostrando así la importancia de considerar evaluaciones con flujos de trabajo.

7 Conclusiones

En este trabajo se propone el uso de técnicas de minería de procesos sobre trazas de sistemas de procesamiento distribuido para descubrir flujos de trabajo que puedan ser consideradas posteriormente en pruebas de rendimiento de dichos sistemas. Se ha evaluado la propuesta con la aplicación “State-Machine Simulator and Minera” en las trazas disponibles de un clúster de Google, obteniendo de los resultados el descubrimiento de flujos de trabajo que reflejan probables relaciones de dependencia entre los trabajos, las cuales podrían afectar las pruebas de rendimiento si es que no son tomadas en cuenta al momento de realizar las evaluaciones. La metodología propuesta puede ser aplicada por investigadores para obtener flujos de trabajos distribuidos realistas, los cuales pueden ser aprovechados para realizar evaluaciones más efectivas a proyectos de investigación sobre plataformas de procesamiento distribuido.

Referencias

1. Van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. Knowledge and Data Engineering,IEEE Transactions on 16(9), 1128–1142 (2004)
2. Van der Aalst, W.M., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.: Workflow mining: a survey of issues and approaches. Data & knowledge engineering 47(2), 237–267 (2003)

3. Abad, C.L., Lu, Y., Campbell, R.H.: DARE: Adaptive data replication for efficient cluster scheduling. In: Proceedings of the International Conference on Cluster Computing (CLUSTER) (2011)

4. Agrawal, R., Gunopulos, D., Leymann, F.: Mining process models from workflowlogs. Springer (1998)

5. Ananthanarayanan, G., Agarwal, S., Kandula, S., Greenberg, A., Stoica, I., Harlan, D., Harris, E.: Scarlett: Coping with skewed popularity content in MapReduce clusters. In: Proceedings of the European Conference on Computing Systems (EuroSys) (2011)

6. Ananthanarayanan,G., Ghodsi, A., Wang, A., Borthakur, D., Kandula, S., Shenker,S., Stoica, I.: PACMan: Coordinated memorycaching for parallel jobs. In: Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI) (2012)

7. Apache Hadoop. `hadoop.apache.org` (Apr 2013), `hadoop.apache.org`, last accessed: April 15, 2013

8. Burattin, A.: Heuristics miner for time interval. In: Process Mining Techniques inBusiness Environments, pp. 85–95. Springer (2015)

9. Chen, Y., Ganapathi, A., Griffith, R., Katz, R.: The case for evaluating mapreduce performance using workload suites. In: Modeling, Analysis & Simulation of Computerand TelecommunicationSystems (MASCOTS), 2011IEEE 19th International Symposium on. pp. 390–399. IEEE (2011)

10. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008)

11. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H., Weijters, A., Van Der Aalst,W.M.: The prom framework: A new era in process mining tool support. In: Applications and Theory of Petri Nets 2005, pp. 444–454. Springer (2005)

12. Gu, T., Zuo, C., Liao, Q., Yang, Y., Li, T.: Improving mapreduce performance bydata prefetching in heterogeneous or shared environments. International Journal of Grid & Distributed Computing 6(5) (2013)

13. Isard, M., Budiu, M., Yu, Y., Birrell, A., Fetterly, D.: Dryad: Distributed dataparallel programs from sequential building blocks. In: Proceedings of the European Conference on Computing Systems (EuroSys) (2007)

14. Lou, J.G., Fu, Q., Yang, S., Li, J., Wu, B.: Mining program workflow from interleaved traces. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 613–622. ACM (2010)

15. Olston, C., Reed, B., Srivastava, U., Kumar, R., Tomkins, A.: Pig Latin: A notso-foreign language for data processing. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (2008)

16. Reiss, C., Tumanov, A., Ganger, G.R., Katz, R.H., Kozuch, M.A.: Heterogeneityand dynamicity of clouds at scale: Google trace analysis. In: Proceedings of the ACM Symposium on Cloud Computing (SoCC) (Oct 2012)

17. Reiss, C., Wilkes, J., Hellerstein, J.L.: Google cluster-usage traces: format + schema. Technical report, Google Inc., Mountain View, CA, USA (Nov 2011), revised 2012.03.20. Posted at http://code.google.com/p/googleclusterdata/wiki/TraceVersion2

18. Reiss, C., Wilkes, J., Hellerstein, J.L.: Obfuscatoryobscanturism:makingworkloadtraces of commercially-sensitive systems safe to release. In: Proc. Intl. Workshop on Cloud Management (CLOUDMAN) (Apr 2012)

19. Thusoo, A., Sarma, J., Jain, N., Shao, Z., Chakka, P., Anthony, S., Liu, H., Wyckoff, P., Murthy, R.: Hive: A warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment 2 (2009)

20. Weijters, A., van Der Aalst, W.M., De Medeiros, A.A.: Process mining with theheuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP 166, 1–34 (2006)

21. Wen, L., Wang, J., van der Aalst, W.M., Huang, B., Sun, J.: A novel approach for process mining based on event types. Journal of Intelligent Information Systems 32(2), 163–190 (2009)

22. Wilkes, J.: More Google cluster data. Google research blog (Nov 2011), posted athttp://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html.

23. Zaharia, M., Borthakur, D., Sen Sarma, J., Elmeleegy, K., Shenker, S., Stoica, I.: Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In: Proceedings of the European Conference on Computing Systems (EuroSys) (2010)

24. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: clustercomputing with working sets. In: Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. vol. 10, p. 10 (2010)

25. Zeng, Q., Sun, S.X., Duan, H., Liu, C., Wang, H.: Cross-organizational collaborative workflow mining from a multi-source log. Decision Support Systems 54(3), 1280–1301 (2013)