

Técnicas de ejecución y monitoreo de procesos en Cloud BPM

José Martínez Garro¹, Patricia Bazán²

¹Facultad de Informática UNLP ²LINTI Facultad de Informática UNLP

josemartinezarro@gmail.com, pbaz@ada.info.unlp.edu.ar

Resumen

La entrada de BPM (*Business Process Management*) en el *cloud* ha cambiado varios conceptos asociados a procesos de negocio, especialmente aquellos relacionados con descomposición de procesos, ejecución y monitoreo. El concepto de descomposición ha sido introducido en BPM permitiendo que los procesos sean ejecutados tanto en un ambiente *cloud* como en uno embebido. Esta situación saca ventaja de conceptos tales como sensibilidad de datos, performance de alto cómputo y portabilidad de aplicaciones. Una vez que los procesos han sido descompuestos, los mismos deben ser monitoreados, conservando la perspectiva del modelo original. A su vez, las actividades de los procesos ejecutan transiciones que generan cambios de estado con datos asociados. En el presente trabajo se analizan conceptos asociados a la descomposición de procesos, así como al monitoreo de las actividades y el uso de los objetos de datos generados a través de sus transiciones. A su vez se enuncian las arquitecturas y metodologías implementadas para el desarrollo de estos conceptos.

Palabras clave: *descomposición, monitoreo, OLC, Cloud, BPM.*

Contexto

El presente artículo se enmarca dentro de un proyecto de investigación acerca de

BPM, mejora continua de procesos, entornos de ejecución y su disposición en un entorno de *cloud computing*, encarado dentro del laboratorio LINTI en la Facultad de Informática de la Universidad Nacional de La Plata.

Introducción

La protección de la privacidad es una de las barreras para ejecutar BPM en un ambiente de *cloud*. No todos los usuarios desean poner sus datos sensibles fuera de la organización. Además, es necesario observar la portabilidad de los productos y sus versiones, así como su disponibilidad en un sistema de *cloud*.

Otro aspecto de interés es la eficiencia. Las actividades de cómputo intensivo se pueden beneficiar en la nube debido a la escalabilidad y a la alta disponibilidad de fuerza de cómputo. Las tareas que no sean de cómputo intensivo, por otra parte, no siempre se benefician en este entorno. La performance de una actividad de este tipo que está corriendo en un entorno embebido debería ser mayor que en la nube debido a los datos que deben ser transferidos para poder ejecutar la actividad. Estas actividades pueden a la vez resultar costosas al ser la transferencia de datos uno de los criterios de facturación del *cloud* [1] [2].

Descomposición de procesos

Existe en la bibliografía actual un modelo de distribución de procesos

llamado PAD (Proceso-Actividad-Datos), el cual describe las posibilidades de separación de los componentes de un BPMS entre nodos *cloud* y embebidos alternativamente. Es posible generalizar la distribución de procesos e identificar un patrón en el cual el motor de procesos, las actividades y los datos se despliegan tanto en la nube como en el usuario final. Esta solución presenta dos beneficios potenciales:

1) El motor de procesos regula los flujos de control y de datos. Una actividad recibe datos del motor de procesos y luego de su ejecución, los resultados producidos son consumidos nuevamente. Consideremos ahora una secuencia de actividades que se ubica en la nube, mientras que el motor de procesos se despliega en el usuario final. Debido a que la transferencia de datos es uno de los criterios de facturación del modelo en cuestión, estas situaciones pueden volverse más costosas cuando se transfieren grandes cantidades de datos entre actividades. Si las actividades se localizan en el *cloud*, los datos son regulados por el motor de procesos situados en él, lo cual reduce la cantidad de datos a ser transferidos entre este y el sistema embebido [1] [3] [4].

2) Cuando la nube no se encuentra accesible, los usuarios pueden ejecutar los procesos de negocio en forma completa en el sistema embebido hasta que el primero vuelva a estar disponible. Para ejecutar un proceso de negocio en un entorno distribuido híbrido (combinación de nodos embebidos y *cloud*), el mismo debe ser dividido en tantas partes como servidores intervinientes existan. Resultaría conveniente para los usuarios tomar una lista de distribución del proceso y sus actividades. La comunicación entre los

nodos involucrados (embebidos o no) puede ser descripta utilizando un lenguaje de orquestación, como BPEL. El monitoreo de procesos de negocio es más complicado ahora, debido a que el proceso ha sido dividido en dos o más partes. Como solución, una herramienta de monitoreo puede ser desarrollada para el proceso original, a través de la combinación de detalles que surgen del monitoreo de partes individuales [1] [2] [5] [6].

Monitoreo de procesos en un esquema híbrido

Tal como se ha enunciado con anterioridad, los mayores problemas de utilizar un modelo de proceso particionado son, además de su ejecución, la recolección y el monitoreo de las diferentes instancias distribuidas, y a su vez la visualización de las mismas bajo la óptica del modelo original al que estas pertenecen. Para enfrentar este desafío se ha diseñado una solución que considera componentes distribuidos e intercomunicados que forman una arquitectura. Por un lado es necesario asociar las diferentes instancias de proceso iniciadas en forma de cadena, con el propósito de recolectar información de las mismas mediante el acceso a los servidores involucrados. De esta manera, cuando una instancia finaliza en un servidor, esta inicia automáticamente una nueva instancia correspondiente a la partición de proceso siguiente, en función del modelo de distribución. Si bien existen distintas maneras de implementar la unión del flujo de ejecución en instancias distribuidas, en nuestro caso hemos optado por utilizar *Bonita Open Solution* [15] como BPMS, junto con sus conectores. Hemos utilizado esta tecnología para implementar los modelos de proceso particionados, y los conectores como mecanismo para iniciar

instancias en nuevos servidores, mediante el uso de clases Java. De esta manera, cada instancia al finalizar ejecuta un conector que permite iniciar un nuevo caso en el siguiente servidor distribuido de la arquitectura, enlazando automáticamente la traza de ejecución de los procesos [7].

Se ha construido a su vez una aplicación de monitoreo encargada de recolectar datos desde los distintos nodos de la arquitectura, así como de mostrarlos en forma integrada de acuerdo a la perspectiva del modelo original descompuesto. Para acceder a todos los nodos involucrados, resulta de importancia poseer un catálogo con los servidores existentes y su información de localización actualizada. Cada uno de estos servidores tiene una copia de un servicio web (*getInstanceService*), el cual recibe un *id* de definición de proceso y retorna información de cada instancia existente en el servidor, asociada con la definición enviada como parámetro. La información retornada incluye *id* de instancia, estado actual (en ejecución, completada, suspendida), actividad actual si la instancia no se encuentra aún finalizada, fecha de inicio y fin. De esta manera la aplicación envía invocaciones del servicio web a cada uno de los servidores distribuidos, concatenando las respuestas obtenidas y mostrándolas en forma integrada. Luego, es posible obtener información detallada de las actividades que componen cada instancia. Para esto se dispone de otro servicio web (*getInstanceActivityService*), el cual retorna información asociada a la instancia enviada: *id* de actividad, participante, fecha de inicio, estado actual y fecha de fin. Una vez finalizada la etapa de recolección, la aplicación web se encargará de concatenar los datos y mostrarlos de manera integrada y con transparencia sobre su ubicación. En la Figura 1 podemos observar la arquitectura

de la aplicación mencionada, con cada uno de los componentes intervinientes [1] [3] [7] [9] [14].

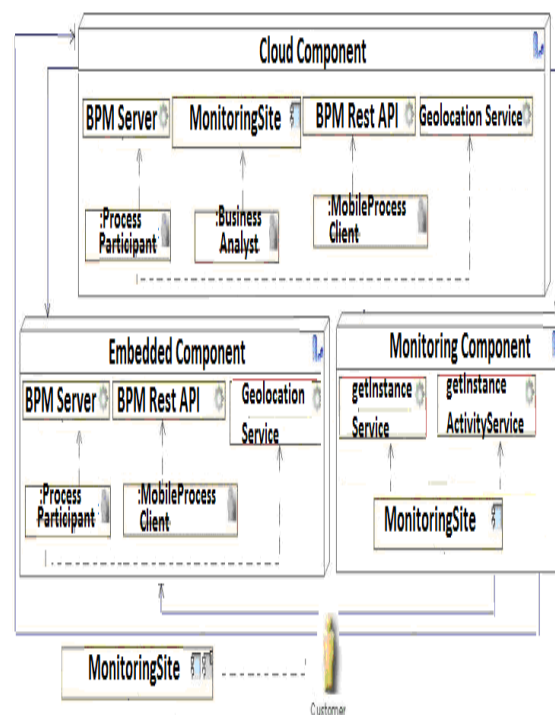


Fig 1: Arquitectura de la aplicación de monitoreo [14]

Uso de objetos de datos OLC en el monitoreo de procesos

En el campo de BPM, el monitoreo se utiliza para observar el comportamiento de las instancias de proceso, así como para reaccionar a eventos y predecir los futuros pasos del proceso en ejecución. Los procesos que son automatizados usando sistemas de información (por ejemplo motores de proceso), pueden ser correctamente monitoreados dado que dichos sistemas frecuentemente ofrecen capacidades de registro, siendo así los procesos fácilmente reconocidos. En contraste, en ambientes donde los procesos deben ser ejecutados manualmente en una gran porción, como por ejemplo en una dependencia de cuidado de la salud, un gran número de eventos no

son capturados en forma automática. De esta manera, la información acerca de los eventos de proceso se encuentra incompleta, y por lo tanto es realmente difícil establecer proposiciones sobre el progreso del proceso mediante la aplicación del concepto de puntos de monitoreo. Un punto de monitoreo de eventos está relacionado a ciertos eventos capturados por un sistema IT conectado a una fuente específica de eventos, por ejemplo una base de datos o un lector de códigos de barra, e informa cuando ciertas transiciones de estado (por ejemplo habilitado, iniciado o finalizado) ocurren en una actividad. En este caso, “probabilístico” significa que es posible proveer un índice indicativo acerca del progreso del proceso, pero el mismo es solo una aproximación. En términos de implementación, la aplicación de monitoreo antes mencionada debe ser capaz de recolectar la información de estado del proceso, y al mismo tiempo, recuperar la información marcada como “observable” en las transiciones del mismo. Para esto es necesario enlazar la información en la base de datos del motor de procesos con la información relativa a los distintos objetos OLC distribuidos e identificados durante la etapa de diseño [2] [8] [10].

En la Figura 2 podemos ver como las diferentes transiciones insertan información en el almacén de eventos distribuido en tiempo de ejecución. Además vemos que los servicios interactúan con los servidores en forma distribuida. Es importante enlazar la información grabada con los detalles del proceso, de manera de obtener todos los eventos asociados

una vez que la aplicación de monitoreo recolecta los datos a mostrar [11] [12] [13].

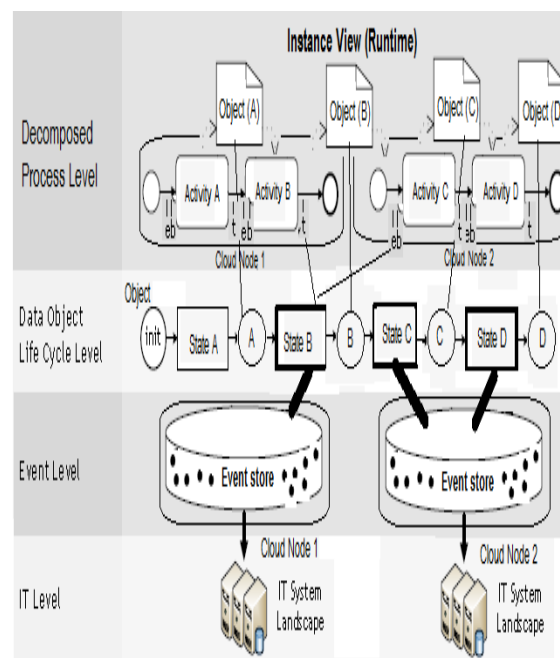


Fig 2: Vista del modelo descompuesto en ejecución

Los servicios web de recolección de datos antes mencionados deben ser modificados de manera tal de considerar ahora, en adición, los objetos de datos OLC que hayan sido marcados como observables. De esta manera, la integración entre los procesos de negocio y la lógica de los sistemas de la organización se realiza a través de los datos distribuidos, y mediante la recuperación de los mismos se pueden mejorar las funciones de monitoreo.

Líneas de Investigación, Desarrollo e Innovación

En la actualidad las líneas de investigación que nos ocupan se centran en la resignificación de conceptos de ejecución de los BPMS embebidos hacia un ambiente de *cloud*: acerca de cómo

tiene mayor sentido en este nuevo contexto la consideración de procesos descompuestos que aprovechen las ventajas de un esquema híbrido, así como el monitoreo de los mismos y el uso de objetos de datos asociados al ciclo de vida. En todos estos ítems cobra una importancia trascendental el uso de mecanismos de seguridad que permitan la ejecución y monitoreo de procesos con estándares similares a los anteriormente considerados en un esquema embebido.

Resultados y Objetivos

Los resultados de la presente línea de investigación han arrojado un análisis de la bibliografía actual sobre BPM en el *cloud* y su adaptación a sistemas híbridos, así como implementaciones de aplicaciones capaces de ejecutar y monitorear procesos distribuidos, con el complemento de los objetos OLC. En la actualidad nos enfocamos en mejorar la aplicación de monitoreo con el fin de considerar los objetos OLC en forma estándar, para no depender de la lógica de la organización particular en su mecanismo de visualización. A su vez, en forma asociada se continúan con trabajos relacionados a *Process Mining* y mejora continua de procesos de negocio.

Formación de Recursos Humanos

BPM ha cobrado importancia dentro del área de tecnología informática, la cual en los últimos años ha evolucionado desde el concepto producto hacia el paradigma de servicios y soluciones. El presente artículo ilustra una línea de investigación iniciada en el año 2008, donde no solo se cubre el desarrollo de procesos BPM sino también la mejora continua de los mismos y la ampliación de sus ambientes de ejecución. En dicha línea se están formando alumnos para

desarrollar su tesina e interactuar con docentes e investigadores formados, incorporando BPM y sus herramientas de soporte como línea de acción para la solución de problemas reales.

Referencias

- [1] T. Kirkham, S. Winfield, T. Haberecht, J. Müller, G. De Angelis, "The Challenge of Dynamic Services in Business Process Management", University of Nottingham, United Kingdom, Springer, 2011
- [2] M. Minor, R. Bergmann, S. Görg, "Adaptive Workflow Management in the Cloud – Towards a Novel Platform as a Service", Business Information Systems II, University of Trier, Alemania, 2012
- [3] M. Mevius, R. Stephan, P. Wiedmann, "Innovative Approach for Agile BPM", eKNOW 2013: The Fifth International Conference on Information, Process, and Knowledge Management, 2013.
- [4] H. Sakai, K. Amasaka. "Creating a Business Process Monitoring System "A-IOMS" for Software Development". Chinese Business Review, ISSN 1537-1506 Junio 2012, Vol. 11, No. 6, 588-595.
- [5] E. Duipmans, Dr. L. Ferreira Pires, "Business Process Management in the cloud: Business Process as a Service (BPaaS)", University of Twente, Abril, 2012.
- [6] JP. Friedenstab, C. Janieschy, M. Matzner, O. Mullerz. "Extending BPMN for Business Activity Monitoring". University of Liechtenstein, Hilti Chair of Business Process Management, Vaduz, Liechtenstein. September 2011.
- [7] N. Herzberg, A. Meyer "Improving Process Monitoring and Progress Prediction with Data State Transition Events". Hasso Plattner Institute at the University of Potsdam. Mayo 2013.
- [8] M. Reichert, J. Kolb, R. Bobrik, T. Bauer. "Enabling Personalized Visualization of Large Business Processes through Parameterizable Views". Hochschule Neu-Ulm, Neu-Ulm, Germany. Noviembre 2011.
- [9] J. Kolar, T. Pitner, "Agile BPM in the age of Cloud technologies", Scalable Computing: Practice and Experience, 2012.
- [10] A. Lehmann and D. Fahland, "Information Flow Security for Business Process Models - just one click away", University of Rostock, Alemania, 2012.
- [11] R. Accorsi, T. Stocker, G. Müller, "On the Exploitation of Process Mining for Security Audits: The Process Discovery Case", Department of Telematics, University of Freiburg, Alemania, 2012.
- [12] S. Zugal, J. Pinggera and B. Weber. "Toward enhanced life-cycle support for declarative processes". JOURNAL OF SOFTWARE: EVOLUTION Marzo 2012
- [13] J. Martínez Garro, P. Bazan "Constructing hybrid architectures and dynamic services in Cloud BPM" Science and Information Conference 2013 Octubre 7-9, 2013 | Londres, RU.
- [14] J. Martínez Garro, P. Bazan "Constructing and monitoring processes in BPM using hybrid architectures". IJACSA Journal, 2014 Enero. Londres, RU.
- [15] Bonita Open Solution <http://es.bonitasoft.com/>. Octubre, 2013.