

CALIDAD DE SERVICIO EN APLICACIONES GRID: UNA REVISIÓN SISTEMÁTICA DE LA LITERATURA

Natalia Trejo, Sandra Casas a
natalia.trejo@gmail.com, scasas@unpa.edu.ar

Unidad Académica Río Gallegos – Universidad Nacional de la Patagonia Austral
Lisandro de La Torre 1070 – 0054-2966-4423113 – Río Gallegos – Santa Cruz – Argentina
Febrero 2012

RESUMEN

La Revisión Sistemática de la Literatura es necesaria toda vez que un investigador se enfrenta con el desafío de resolver un problema que es de su interés. Es un método de investigación consistente en tres etapas: planificación de la revisión, desarrollo de la revisión y publicación de resultados. El presente trabajo es resultado de la ejecución de la planificación de la revisión que está relacionada con la calidad de servicio (Quality of Service, QoS) o requerimientos no funcionales aplicados en etapas tempranas del desarrollo de aplicaciones para Grid computing. De esta forma se presenta el estado del arte acerca del tema antes mencionado.

Palabras claves: calidad de servicio, workflow Grid, aplicación Grid, requisito no funcional, servicio Grid, composición de servicios

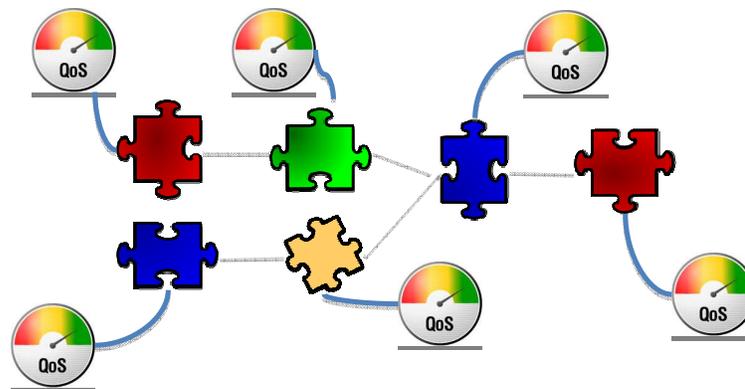
INTRODUCCIÓN

La tecnología Grid (Foster et al., 2001) proporciona un entorno de cómputo distribuido basado en la agregación y compartición segura y coordinada de recursos heterogéneos. Dichos recursos provienen de diferentes organizaciones dinámicamente agrupadas en las llamadas Organizaciones Virtuales (Virtual Organizations, VOs).

Las aplicaciones Grid para sistemas basados en servicios no suelen basarse en un solo servicio, sino que se componen de varios servicios que trabajan juntos de una forma específica para la aplicación con el fin de lograr un objetivo global. El desarrollador de aplicaciones Grid tiene que decidir qué servicios ofrecidos por el Grid deben utilizarse en la aplicación, y especificar los flujos de datos y de control entre ellos. De esta forma, se define un workflow como la automatización de estos flujos en las aplicaciones Grid.

En el trabajo de (Menascé, 2002) se define la Calidad de Servicio (Quality of Service, QoS) como la “combinación de varias cualidades o propiedades de un servicio”. En el contexto de Arquitectura basada en Servicios (Service-oriented Architecture, SOA), es un conjunto de atributos no funcionales que puede influir en la calidad de servicio proporcionada por un servicio Web (W3C Consortium, 2003). Normalmente lo que ocurre es que varios servicios Web o servicios Grid (como extensión de los servicios Web) pueden ejecutar la misma tarea pero con diferentes niveles de calidad. Además los usuarios o aplicaciones pueden tener diferentes expectativas y requisitos. En aplicaciones Grid basadas en servicios Grid, los usuarios construyen los workflows ensamblando servicios Grid que, en muchos casos, ejecutarán tareas complejas con diferentes restricciones de QoS. La Figura 1 representa gráficamente una aplicación Grid como un workflow de servicios Grid, donde cada uno de los ellos puede ejecutarse con diferentes atributos de QoS que a su vez determinarán la QoS con que se ejecute la aplicación Grid completa. Durante las fases tempranas de desarrollo de aplicaciones Grid es importante considerar atributos no funcionales de la aplicación previos a la fase de ejecución, para satisfacer las necesidades cambiantes que pueden tener los servicios o clientes que consumen el servicio Grid.

Figura 1. Representación gráfica de la composición de servicios con QoS



Los diferentes enfoques de desarrollo de software utilizados para construir estas aplicaciones no consideran todas las etapas del ciclo de vida básico del software (Trejo y Casas, 2011). Algunos se centran en la programación, otros en análisis, ninguno en mantenimiento, y sólo al-

gunos toman en consideración los requerimientos no funcionales. Entonces, surge la necesidad de utilizar un paradigma de Separación Avanzada de Concerns (Advanced Separation of Concerns, ASoC) (Brichau et al., 2002) que considere los requisitos funcionales y no funcionales desde el comienzo del desarrollo de estas aplicaciones y también la alta variabilidad de las combinaciones de estos requisitos. Así, este hecho constituye un tema de investigación relevante para las aplicaciones basadas en workflows de servicios Grid.

Por ello, es preciso realizar una Revisión Sistemática de la Literatura (RSL) (Kitchenham, 2004) con el fin de aplicar un método científico que tenga como objeto la problemática antes descrita. Existen diversas metodologías, como por ejemplo (Eelco, 2010), (Caro et al., 2005), entre otras. Particularmente, los autores de (Caro et al., 2005) extendieron la metodología propuesta por Kitchenham y en el presente trabajo se exponen los resultados de la aplicación de esta metodología extendida de RSL. Específicamente se presenta un análisis comparativo acerca de la incorporación de requisitos no funcionales en fases tempranas del desarrollo de aplicaciones para entornos Grid, especialmente las basadas en workflows de servicios Grid.

La estructura del documento es como sigue: la Sección 1 describe la fase de planificación de la RSL. En la Sección 2 se presentan los resultados obtenidos de ejecutar la búsqueda de estudios primarios y se presenta cómo se ha ejecutado la selección de los estudios primarios. En la Sección 3 se resumen brevemente los estudios analizados en la presente RSL. La siguiente sección establece los criterios de comparación que se aplicarán y presenta un análisis comparativo consistente en las respuestas a los interrogantes de investigación planteados en la etapa 1 de la planificación de la RSL. Finalmente se presentan las conclusiones y futuros trabajos.

1 PLANIFICACIÓN DE LA REVISIÓN

1.1. IDENTIFICACIÓN DE LA NECESIDAD DE REVISIÓN

El propósito de la presente RSL consiste en resumir la evidencia existente en relación a cómo se han incorporado los requisitos no funcionales en fases tempranas del desarrollo de aplicaciones para entornos Grid, especialmente las basadas en workflows de servicios. Los interrogantes de investigación son los siguientes:

- ¿Cuáles son los problemas asociados a la no inclusión de requerimientos no funcionales en etapas tempranas del desarrollo de aplicaciones Grid? ó ¿Cuáles es la necesidad de incluir requisitos no funcionales en etapas tempranas de desarrollo de software?
- ¿Cuáles son los requerimientos no funcionales más relevantes asociados a las aplicaciones Grid?
- ¿Qué enfoques de SoC se aplicaron para incluir requerimientos no funcionales en el desarrollo de aplicaciones Grid?
- ¿Qué métodos y/o herramientas existen para incorporar y/o evaluar los requerimientos no funcionales en el desarrollo de aplicaciones Grid?

Los recursos para responder a los interrogantes de investigación anteriores, en principio, son los siguientes:

- Actas de IEEE/ACM International Conference on Grid Computing desde 2000-2010

- Revistas electrónicas y libros disponibles en la biblioteca de la Universidad Nacional de la Plata (UNLP) y en la Universidad Nacional de la Patagonia Austral (UNPA), incluidos los de formato electrónico
- Bibliotecas Digitales, como por ejemplo: Biblioteca Electrónica de Ciencia y Tecnología, Citeseer, etc.
- Trabajos de investigación y Tesis relacionadas con el tema de investigación
- Internet, a través del uso de motores de búsqueda como, por ejemplo, Google Scholar. 2000-2010
- ACM Digital Library como revista electrónica de acceso privado a través de suscripción

1.2. DEFINICIÓN DEL PROTOCOLO DE BÚSQUEDA

1.2.1 Definición del protocolo de búsqueda

Los términos a emplearse para realizar la búsqueda son compuestos: “Quality of service”, QoS, “QoS constraint”, “QoS restriction”, “non-functional requirements”, “Grid service”, “Grid workflow”, “Grid application”, “service composition”.

1.2.2 Términos compuestos y Combinaciones

Las combinaciones de términos más relevantes para el tema de la RSL son: QoS + “Grid workflow”, “QoS constraint” + “Grid service”, QoS + “Grid service”, “QoS constraint” + “Grid workflow”, “non-functional requirement” + “Grid application”, “non-functional requirement” + “Grid service”, “service composition” + QoS + Grid

1.2.3 Estrategias de búsquedas

Para la búsqueda de información, se consideran los artículos publicados en los últimos diez años, considerando la ocurrencia de palabras clave en títulos, resúmenes y palabras clave. A continuación se definen las normas que seguirá la investigación respecto del proceso de búsqueda en las fuentes de información definidas en el apartado anterior.

- **En recursos con herramientas de búsqueda:** Ingresar en forma escalada los términos y/o combinaciones de ellos.
- **En CD's de actas de congresos:** Buscar sobre los títulos, en los índices, y luego sobre los resúmenes y palabras claves. Conjuntamente usar la función de búsqueda de la aplicación.
- **En textos no electrónicos:** Revisión manual de título, resumen, palabras clave.
- **En Internet:** Acceso a documentos mediante motores de búsqueda como Scopus, Google Scholar, CiteSeer, entre otros. En caso que los documentos no estén accesibles, buscar en las páginas de sus autores o en sitios de publicación alternativos.
- **Autores:** Identificar a los autores relevantes, acceder directamente a sus páginas personales para la búsqueda de material.

- **En artículos:** Detectar referencias bibliográficas de utilidad, en base a ellas buscar directamente el documento citado usando los antecedentes que aparecen en la citación (autor, título, conferencia, etc.).

1.2.4 Registro de resultados

Los resultados de las búsquedas y el origen de los mismos se registran mediante tablas que resumen los resultados de manera general, según la combinación de los términos utilizados para realizarla. Asimismo se registran las referencias y los trabajos descargados mediante la herramienta de gestión de referencias Papers2 para permitir acceder al material y a las referencias sin necesidad de conexión a Internet. La extensión del navegador web Mozilla Firefox, Zotero, se utiliza para gestionar material de interés cuando se disponga de conexión a internet y como paso previo a registrar el material en Papers2. También se utiliza un gestor de referencias liviano basado en web con el objetivo de ser utilizado como repositorio de referencias en formato bibtex para facilitar el acceso de varios investigadores.

1.3. DEFINICIÓN DEL PROTOCOLO DE REVISIÓN

En esta sección se especifican los métodos que usados para emprender la revisión sistemática.

1.3.1 Normas de revisión

Se incluyen aquellos trabajos que abordan los requerimientos no funcionales en aplicaciones Grid y que se enmarquen en algunos de los siguientes tópicos:

- Planteamiento/Análisis del problema de la no inclusión de requerimientos no funcionales en etapas tempranas del desarrollo de aplicaciones Grid. Planteamiento de la necesidad de incluir atributos no funcionales en el desarrollo de aplicaciones Grid.
- Análisis de las características específicas de aplicaciones Grid relacionadas con requerimientos no funcionales.
- Propuesta de enfoques de ASoC que incluyan y/o evalúen requerimientos no funcionales en el desarrollo de aplicaciones Grid.
- Herramientas para incorporar y/o evaluar QoS/requerimientos no funcionales en aplicaciones Grid.

1.3.2 Criterios de inclusión

Se excluyen aquellos estudios que a pesar de contener los términos de búsqueda o combinación de ellos, no contengan información relevante sobre el tema y/o no aborden los tópicos de interés, por ejemplo: “QoS y web services” ó “QoS y Grid scheduler”.

1.3.3 Criterios de exclusión

Si no se dispone del trabajo completo, se consigue para asegurar que tiene relevancia para el tópico de investigación. Leer al menos el resumen e introducción y decidir si incluir o no el estudio en base a los criterios de este protocolo. En la copia digital del trabajo, adjuntar comentarios sobre el mismo documento en un lugar visible. Se lleva un control acerca de los estudios primarios aceptados y rechazados, registrando el motivo de exclusión.

1.3.4 Estrategia de extracción de datos

Por cada estudio seleccionado, se realiza una lectura, en 2 etapas, con el objeto de extraer datos para el trabajo.

Etapla 1. En esta etapa se debe leer el resumen, introducción, trabajos relacionados, conclusión y referencias. Donde podremos obtener la información siguiente:

- A qué comunidad está orientado el artículo. [Introducción, Trabajos relacionados, Referencias]
- Cuáles son sus contribuciones (según los autores). [Resumen, Introducción, Conclusión]
- Cuáles son las posibles consecuencias de las contribuciones (Aplicaciones directas, nuevas técnicas, nuevas áreas de investigación, etc.).[Introducción].

Etapla 2. Se leen los preliminares y el cuerpo del artículo, esto permite:

- Incluir en forma detallada la información que necesitemos para nuestra revisión.
- Comprender un experimento, los fundamentos de un marco de trabajo, las características de un modelo, etc.

1.3.5 Estrategia de síntesis de datos

Los datos son sintetizados de acuerdo a los siguientes temas:

- Problemas de la no inclusión de requerimientos no funcionales en workflows ó aplicaciones Grid en etapas tempranas de su desarrollo.
- Principales requerimientos no funcionales relacionados con workflows ó aplicaciones Grid
- Propuestas de enfoques de ASoC para incorporar/evaluar requerimientos no funcionales en workflows ó aplicaciones Grid
- Herramientas para incorporación/evaluación de requerimientos no funcionales en etapas tempranas del desarrollo de workflows ó aplicaciones Grid.

2 DESARROLLO DE LA REVISIÓN

2.1. Búsqueda de Estudios Primarios

La búsqueda de estudios primarios se realizó en base al protocolo de búsqueda definido en la planificación de la RSL y sobre los recursos de información disponibles. Este protocolo sufrió una modificación en cuanto a los términos que se utilizaron para realizar la búsqueda. Se usaron combinaciones de términos para realizar la búsqueda ya que los términos por sí solos, es decir, sin realizar las combinaciones, arrojan gran cantidad de resultados heterogéneos y alejados del tema de la RSL. La Tabla 1 muestra el resultado de la búsqueda en fuentes que se acceden desde Internet.

Tabla 1: Búsquedas primarias

Términos	Citeseer	IEEE	Springer Link	Science Direct	Open Thesis	Google Scholar	ACM
QoS + “Grid computing”	422	3467	85	459	129	6960	3708
QoS + “Grid service”	239	962	416	214	37	2500	3707
QoS + “Grid workflow”	32	240	59	29	7	684	809
"Non-functional requirement” + “Grid application”	13	1024	0	7	2	12	1
QoS +Grid + “service composition”	73	298	973	213	188	1950	947

2.2 Selección de Estudios Primarios

La selección de los estudios se hizo en base al protocolo de revisión definido. Con los estudios seleccionados se procedió a la extracción de datos y en los no seleccionados se registró la información relativa al motivo del rechazo. Este registro se hizo agregando notas en el documento pdf y registrando el motivo en el software Papers2.

3 ANÁLISIS

En esta etapa se extrajo la información de interés de los estudios siguiendo el procedimiento del protocolo de revisión. Se escribió un resumen de cada estudio. Además se registró la información necesaria para la gestión, como la relativa a la bibliografía utilizando el software Papers2 para mantener los datos más relevantes de cada estudio revisado. En cada artículo se registró sobre su impreso o versión digital, el contenido más relevante a la investigación. A continuación se presenta el resumen de cada uno de los estudios seleccionados.

3.1. QOS-DRIVEN SERVICE CONFIGURATION IN COMPUTATIONAL GRIDS

El trabajo de (Musunoori et al., 2005) presenta un enfoque de configuración de servicios con soporte para QoS. De esta forma, se permite a los desarrolladores separar la especificación funcional de los requisitos de QoS a partir de decisiones de implementación que dependen del entorno de despliegue de Grid.

Los autores plantean que no incluir requisitos de QoS a la aplicación Grid puede resultar en una baja performance de ésta. No definir o gestionar los requisitos de QoS implica que se asuma una configuración particular de una plataforma de despliegue, que producirá un desempeño inaceptable en otras configuraciones aún con los recursos adecuados. Las aplicaciones Grid se consideran como sensibles a QoS ya que tendrán diferentes desempeño con recursos escasos o si el despliegue no es cuidadosamente configurado antes de la ejecución.

El trabajo no hace alusión a características no funcionales específicas para aplicaciones Grid, el caso de estudio analizado sólo considera la latencia y la precisión de los resultados obtenidos de la aplicación Grid.

El enfoque de ASoC que se propone es el desarrollo de software basado en componentes (Component-based Software Development, CBSD). Se define un servicio de planificación que gestiona los requisitos de QoS y que puede ser conectado al resto de los servicios que conforman el workflow Grid.

Los autores proponen un framework que incluye al servicio planificador y modelos para resolver programación multi-objetivo al demandar requisitos de QoS en múltiples dimensiones.

3.2. A MODEL-DRIVEN APPROACH FOR GRID SERVICES ENGINEERING

Los autores de (Manset et al., 2005) presentan una técnica para especificar aplicaciones Grid usando el enfoque de desarrollo basado en modelos. Estos modelos se transforman para automatizar la adaptación a plataformas específicas incluyendo modelos para requisitos de QoS. Presentan un framework que implementa la combinación del enfoque dirigido por modelos y el enfoque centrado en la arquitectura para el desarrollo de aplicaciones Grid.

Se plantea que el diseño de aplicaciones Grid generalmente es afectado por muchas restricciones de diferentes tipos, introducidas por el desarrollador tales como las de QoS, o que son impuestas de cierta forma por la plataforma de ejecución destino de la aplicación. Esto motiva a que los atributos de QoS deban especificarse desde fases tempranas del desarrollo de aplicaciones Grid.

El trabajo no se realiza una descripción de las características de QoS particulares para aplicaciones Grid.

Los autores aplican el desarrollo dirigido por modelos (Model-driven Software Development, MDSD) para realizar la composición de servicios Grid considerando requisitos de QoS desde el diseño hasta el despliegue del workflow Grid.

Se propone un framework basado en las herramientas ArchWare que incluye un lenguaje y un editor gráfico. Este framework permite a los usuarios describir los servicios y la composición de una manera amigable y transparente para generar de manera semiautomática la aplicación Grid donde se incluyen los requisitos no funcionales.

3.3. HIGH-LEVEL COMPOSITION OF QOS-AWARE GRID WORKFLOWS: AN APPROACH THAT CONSIDERS LOCATION AFFINITY

El trabajo de (Brandic et al., 2006) presenta el desarrollo de un lenguaje que soporta la especificación de restricciones de QoS relacionadas con aspectos de seguridad y legales, además del costo y rendimiento. Se define un lenguaje específico de dominio basado en UML para workflows Grid con soporte de QoS además de extender el editor gráfico llamado Teuta con el lenguaje desarrollado e integrar este editor con el motor de ejecución de workflows llamado QoS-aware Grid Workflow Engine (QWE).

Se plantea la necesidad de que además de requisitos no funcionales como la performance y el costo de ejecución, son importantes los aspectos de seguridad y legales. La tecnología Grid debe garantizarlos desde el momento de la especificación del workflow Grid hasta la ejecución del mismo. Los autores consideran útil que el usuario pueda decidir la ubicación de los recursos Grid donde ciertas actividades serán ejecutadas debido a aspectos de seguridad y legales que deban respetarse.

Se incluyen como atributos de QoS al precio, tiempo de inicio de ejecución, tiempo de final de ejecución y afinidad geográfica referido a la ubicación donde los recursos Grid ejecutarán las tareas representadas por los servicios del workflow Grid.

El trabajo no hace referencia a un enfoque de ASoC específico sin embargo define un lenguaje basado en UML para QoS-aware Grid workflows (QoWL) extendiendo el lenguaje BPEL con elementos para especificar restricciones de QoS. Los autores definen un Lenguaje Específico de Dominio (Domain Specific Language, DSL) para los workflows con soporte para QoS extendiendo los diagramas de actividades de UML.

Se extiende el editor gráfico Teuta para permitir que el usuario defina requerimientos de QoS y validar la especificación generada. Dicha especificación luego es usada por el motor de ejecución QWE.

3.4. QOS-DRIVEN SERVICE COMPOSITION MODELING WITH EXTENDED HIERARCHICAL CPN

El trabajo de (Li et al., 2007) describe técnicas de análisis y modelado para realizar la composición de servicios dirigida por QoS usando una red de Petri coloreada jerárquica, llamada QSC-Nets (QoS-Driven Service Composition Nets). Esta técnica permite que los desarrolladores expresen explícitamente en el modelo las características no funcionales de la composición de servicios.

Consideran que es preciso tener en cuenta las propiedades dinámicas de los recursos donde se ejecutan los servicios del workflow Grid y también permitir que esta selección no sea totalmente manual y rígida. Esto significa que una vez definida por el desarrollador se pueda realizar y gestionar automáticamente si estas propiedades sufren cambios durante la ejecución de workflow.

Los autores consideran como atributos no funcionales a los siguientes: disponibilidad (a través del análisis del máximo número de servicios concurrentes que pueden ejecutarse en un recurso), costo, duración y confiabilidad.

No se propone la utilización de ningún enfoque de ASoC para incluir o evaluar los atributos de QoS definidos por el desarrollador.

No proponen ninguna herramienta sino que sólo describen técnicas y métodos basados en QSC-nets para manejar las propiedades dinámicas de la composición de servicios durante la ejecución del workflow. La estructura del modelo se especifica luego utilizando el lenguaje BPEL4WS.

3.5. ENABLING QOS FOR SERVICE-ORIENTED WORKFLOW ON GRID

Los autores de (Guo et al., 2007) presentan un sistema para la gestión de workflows con soporte para QoS, éste asegura que los workflows cumplan con los requerimientos de QoS predefinidos a la vez que los optimiza.

Ellos plantean la necesidad de incluir a un sistema gestor de workflows, características que permitan aceptar y controlar requisitos de QoS definidos por el usuario, debido a la gran cantidad de componentes dispersos en diferentes sitios que forman parte de un entorno Grid. La naturaleza dinámica de estos recursos en cuanto a disponibilidad y carga de trabajo que soportan, puede resultar en que el usuario no reciba la respuesta esperada durante la ejecución del workflow.

Los autores no realizan una enumeración exhaustiva de los requisitos de QoS específicos para la aplicaciones Grid, sin embargo mencionan algunos como máximo tiempo de respuesta, velocidad de procesamiento de los nodos, confiabilidad. Los requisitos de QoS son divididos en concordancia con los elementos que forman el lenguaje BPEL que utiliza el gestor para generar la especificación del workflow: requerimientos globales, requerimientos para una actividad de invocación, requerimientos para múltiples actividades de invocación y requerimientos separados y diferentes para cada actividad.

Explícitamente no proponen ningún enfoque de ASoC para realizar la composición de servicios.

Se describe la arquitectura del sistema gestor de workflow que recibe del editor un documento BPEL junto con otro documento que apunta a éste último y agrega los requisitos de QoS utilizando la notación XPath de XML para hacer esa referencia.

3.6. SOAG: SERVICE ORIENTED ARCHITECTURED GRIDS Y ADOPTION OF APPLICATION SPECIFIC QOS ATTRIBUTES

El trabajo de (Wahib et al., 2008) presenta la integración de SOA con computación Grid a través de un modelo Grid cuya arquitectura es orientada a Servicios (Service Oriented Architected Grid, SOAG).

Los autores plantean como una necesidad que el desarrollador de aplicaciones Grid pueda definir requisitos de QoS debido a que los mismos, en general, son muy específicos de la aplicación Grid que se está desarrollando.

Se clasifican los atributos de QoS en básicos (que incluyen al costo, tiempo, disponibilidad y fiabilidad) y específicos de la aplicación (definidos por el desarrollador para cada aplicación en particular).

No presentan ningún enfoque de ASoC para incluir o evaluar los requisitos no funcionales en el diseño de aplicaciones Grid.

Se propone un framework donde los desarrolladores, a través de herramientas para entorno de Construcción de Aplicaciones, definen atributos de QoS específicos de la aplicación a través de portlets. Estos atributos pueden ser incluidos en el proceso de planificación de trabajo a través del servicio intermediario de recursos (resource broker) el cual, a través de la interacción con los recursos como servicios, puede incluir atributos de QoS definidos para la aplicación. De la misma forma, se pueden establecer acuerdos de nivel de servicio ó SLA (Service-Level Agreement) de las características de QoS específicas de la aplicación que los usuarios finales pueden negociar con los intermediarios de recursos.

3.7. COST Y ACCURACY SENSITIVE DYNAMIC WORKFLOW COMPOSITION OVER GRID ENVIRONMENTS

Los autores de (Chiu et al., 2008) presentan un framework para incorporar QoS durante la composición de workflows. El framework aplica un algoritmo para seleccionar los recursos que satisfacen los requisitos de QoS y luego aplica un esquema de ajuste de precisión para adaptar el workflow a las restricciones dadas.

Se considera un sistema Grid relacionado con la investigación geoespacial, donde los repositorios de datos con grandes volúmenes de información están conectados mediante redes que tienen un potencial de latencia alta y donde los costos de acceso también lo son. Los autores

plantean que no incluir el tiempo de ejecución y la precisión como requisitos de QoS puede generar resultados erróneos debido a los altos costos de acceso y al movimiento de grandes volúmenes de información en un entorno de alta latencia. La inclusión de requisitos no funcionales permite satisfacer diferentes requerimientos ya que algunos usuarios pueden priorizar el tiempo de respuesta y otros pueden demandar respuestas más precisas.

Las características consideradas como requisitos no funcionales fueron el tiempo de ejecución, la exactitud de los resultados y las limitaciones de ancho de banda de la red.

Aunque no se describe un enfoque de ASoC, los autores proponen que el usuario durante la fase de creación del workflow pueda definir las restricciones antes mencionadas.

El framework propuesto soporta las preferencias del usuario como requisitos de QoS relacionados con el tiempo y la precisión. Incluye modelos para evaluar los costos asociados con la recuperación de cada conjunto de datos o la ejecución de cualquier servicio; toma como modelos de entrada aquellos que predicen la exactitud asociada con los resultados de un determinado servicio, en función de la exactitud de los datos de entrada al servicio. Considerando asimismo las restricciones especificadas (precisión y coste) como entrada, el sistema automáticamente compone el workflow no incluyendo los servicios que no pueden cumplir con las restricciones.

3.8. ADVANCED QOS METHODS FOR GRID WORKFLOWS BASED ON META-NEGOTIATIONS Y SLA-MAPPINGS

El trabajo de (Brandic et al., 2008) presenta una propuesta para incorporar a workflows Grid los requisitos no funcionales utilizando una arquitectura que gestiona lo que llaman meta-negociaciones y mapeo de SLA (Service Level Agreements).

Los autores dan por hecho que debe existir un mecanismo que asegure la calidad del servicio en entornos Grid donde los consumidores y proveedores de servicios se comunican de forma dinámica sin conocimiento previo mutuo. En modelos de compartición de recursos comerciales, la QoS se negocia entre las partes siguiendo protocolos de negociación específicos que se registran en SLA. Generalmente se asume que las partes que se comunican conocen a priori los protocolos y las plantillas de SLA a utilizar. Sin embargo esto no ocurre entre las partes que se comunican utilizando un entorno Grid.

Los requisitos no funcionales que consideran en el trabajo incluyen: tiempo de ejecución, confiabilidad, disponibilidad y costo de servicio.

Aunque no proponen una enfoque de ASoC para incluir y/o evaluar los requisitos no funcionales, sí proponen que en el editor de workflows se incorporen dichas características. Estos mecanismos de meta-negociaciones y mapeos de SLA se incorporan a una herramienta de gestión de workflows Grid, llamada Amadeus y donde el modelado se realiza con el editor UML para workflows llamado Teuta. La representación UML del workflow se traduce a XML siguiendo la sintaxis de QoWL (Quality of Service aware Grid Workflow Language) de manera similar a cómo lo realizaron en (Brandic et al., 2006).

3.9. QOS-BASED DECISION SERVICES IN GRIDS

El trabajo de (Kyriazis et al., 2008) presenta servicios que permiten incorporar la QoS en aplicaciones Grid basadas en servicios. Los servicios de decisión establecen las garantías de QoS y dan soporte para que los requerimientos de la aplicación Grid se cumplan. El servicio componente de mapeo de workflow permite establecer la relación entre los procesos del workflow y los servicios ofrecidos por el Grid asegurando la QoS definida por los parámetros y preferencias del usuario. El servicio de negociación de recursos, a través de SLAs, reconcilia los intereses de los proveedores y los consumidores, y negocia entre los proveedores de servicios y el broker de recursos.

Los autores plantean que los atributos no funcionales son fundamentales en entornos Grid si se desean ejecutar aplicaciones comerciales o de misión crítica en ellos.

Las propiedades de QoS que se mencionan son: tiempo de ejecución, costo y disponibilidad. No se propone ningún enfoque de ASoC para la incorporación de QoS en composición de servicios.

Se ofrecen dos tipos de servicios que permiten definir propiedades no funcionales, sin embargo no se describe cómo se incorporan en algún sistema de gestión de workflows.

3.10. GRID RESOURCES PRICING: A NOVEL FINANCIAL OPTION BASED QUALITY OF SERVICE-PROFIT QUASI-STATIC EQUILIBRIUM MODEL

Los autores de (Allenator y Thulasiram, 2008) presentan la aplicación de teoría financiera para diseñar, desarrollar y simular modelos de costos de recursos Grid para equilibrar las restricciones de QoS. Se considera como activos reales a los recursos Grid con el fin de capturar el valor realista de commodities de cómputos Grid. Luego se valorizan los recursos Grid para resolver el modelo financiero.

Los autores no plantean cuáles inconvenientes produce la no inclusión de requisitos no funcionales en el diseño de una aplicación Grid.

Este trabajo sólo contempla como requisito no funcional al costo del recurso Grid a través de regular el mismo utilizando una infraestructura de políticas de precios basada en SLAs. No se define un enfoque de ASoC para incorporar los requisitos de QoS.

Se propone una infraestructura de políticas de precios para recursos Grid que comprende una capa de servicios Grid, que incluye el middleware, y una capa de optimización de precios situada en una capa superior a la capa de middleware.

3.11. A FRAMEWORK FOR QOS-BASED RESOURCE BROKERING IN GRID COMPUTING

El trabajo de (Ranaldo y Zimeo, 2008) consiste en modificar un framework existente para negociar los recursos en entornos Grid en base a características relacionadas con QoS. El framework automáticamente asigna las tareas que integran la aplicación Grid a los recursos usando una estrategia de emparejamiento (matching) basada en restricciones de costo y tiempo de ejecución.

Los autores plantean que debido a las características intrínsecamente heterogéneas de los recursos en un entorno Grid, las características de QoS son consideradas en tiempo de ejecución de las aplicaciones para Grid computing.

Como atributos de QoS específicos para establecer en aplicaciones para Grid computing, incluyen tiempo de ejecución para las tareas de cómputo, capacidad de almacenamiento mínimo, cantidad de tareas que puede ejecutar el recurso Grid, entre otros.

Los autores no aplican ningún enfoque de ASoC para incluir requerimientos no funcionales, sólo hacen uso de ontologías para describir las atributos de QoS.

Se extiende el framework llamado Service Matchmaker para incluir e integrar mediante la estrategia de matching por minimización de tiempo a los requisitos de QoS relacionados con el tiempo de ejecución, el costo y capacidad del recurso en cuanto al número total de tareas que puede ejecutar.

3.12. IMAGING SERVICES ON THE GRID AS A PRODUCT LINE: REQUIREMENTS Y ARCHITECTURE

El propósito del trabajo de (Acher et al., 2008) consiste en analizar la variabilidad de los aspectos funcionales y no funcionales del dominio de procesamiento de imágenes médicas. Se propone una arquitectura en la cual los servicios se organizan dentro de una arquitectura de línea de productos y donde los metamodelos dan soporte para organizar la información necesaria. La estrategia es presentar una línea de productos de servicios para describir las principales variaciones en las especificaciones funcionales y no funcionales de los servicios, así como en los posibles flujos de información y procesos en un workflow Grid.

Los autores de este trabajo aunque no realizan un análisis específico del problema de la no inclusión de atributos no funcionales en el desarrollo de aplicaciones para Grid computing. Plantean la aplicación del paradigma de Línea de Producto de Software (Software Product Line, SPL) considerando la variabilidad tanto en los requerimientos funcionales como no funcionales de la aplicación bajo estudio.

El workflow de servicios Grid para realizar el procesamiento de imágenes médicas considera fundamentales los atributos de QoS con que se ejecutará el mismo en el entorno Grid de destino. Principalmente definen como importantes los aspectos de QoS relacionados con el tiempo de ejecución, costo de cómputo, confiabilidad, exactitud, precisión o reproducibilidad y previsibilidad.

Los autores proponen una arquitectura para aplica SPL compuesta por un framework (Service Product Line Framework, SPLF) para describir el modelo de dominio, un repositorio de servicios existentes para el dominio del problema y metamodelos que capturan el conocimiento del SPLF.

4 RESULTADOS

4.1. CRITERIOS DE COMPARACIÓN

Los criterios de comparación se relacionan con los interrogantes de investigación que se definieron en la Sección 2.1. A continuación se resumen dichos criterios:

1. Planteamiento del problema de la no inclusión de requerimientos no funcionales en etapas tempranas del desarrollo de aplicaciones Grid. Necesidad de la inclusión de atributos de QoS en fases tempranas del desarrollo de estas aplicaciones.
2. Análisis de las características específicas de aplicaciones Grid relacionadas con requerimientos no funcionales.
3. Propuesta de enfoques de ASoC que incluyan y/o evalúen requerimientos no funcionales en el desarrollo de aplicaciones Grid.
4. Herramientas/metodologías para incorporar y/o evaluar requerimientos no funcionales en aplicaciones Grid.

4.2. EVALUACIÓN

En esta sección se presenta una comparación de la extracción de datos de los trabajos seleccionados, según los criterios establecidos en la sección anterior.

Tabla 2: Problema por no incluir QoS en aplicaciones Grid o necesidad de incluirlos

Referencia	Planteamiento
(Musunoori et al., 2005)	no definir requisitos de QoS en diseño de la aplicación Grid puede resultar en baja performance
(Manset et al., 2005)	se precisa incluirlos para diferenciar la gran cantidad restricciones de diferentes tipos que afectan diseño de aplicación Grid
(Brandic et al., 2006)	se precisa incluirlos porque el usuario no puede decidir la ubicación donde se ejecutarán las tareas
(Li et al., 2007)	se precisa incluirlos para tener en cuenta las propiedades dinámicas de los recursos donde se ejecutan los servicios
(Guo et al., 2007)	se precisa que el usuario pueda seleccionar atributos de QoS debido a la naturaleza dinámica de los recursos
(Wahib et al., 2008)	se precisa incluirlos porque en general son específicos

	de la aplicación
(Chiu et al., 2008)	se precisa incluirlos para evitar resultados erróneos debido a los costos de acceso y al movimiento de grandes volúmenes de información en entorno de alta latencia
(Brandic et al., 2008)	se precisa incluirlos para facilitar la comunicación dinámica sin conocimiento previo mutuo que se produce entre consumidores y proveedores de servicios
(Kyriazis et al., 2008)	se precisa para ejecutar aplicaciones comerciales o de misión crítica en ellos
(Allenator y Thulasiram, 2008)	—
(Ranaldo y Zimeo, 2008)	se precisa debido a las características intrínsecamente heterogéneas de los recursos en un entorno Grid
(Acher et al., 2008)	el usuario puede tener diferentes expectativas en cuanto a resultados de la aplicación: precisión, calidad de los resultados, urgencia de la ejecución, etc

La Tabla 2 resume los resultados relacionados con el criterio 1, que refiere al planteamiento del problema de la no inclusión de características no funcionales en fases tempranas de desarrollo de aplicaciones para Grid computing o la necesidad de incluirlos. Los trabajos de (Li et al., 2007), (Guo et al., 2007), (Ranaldo y Zimeo, 2008), (Manset et al., 2005) y (Brandic et al., 2008) coinciden en que es preciso definir los atributos de QoS debido a las propiedades dinámicas y altamente cambiantes del estado de recursos, de la comunicación de los consumidores y proveedores de estos servicios y de la infraestructura Grid intrínsecamente heterogénea donde se ejecutan las aplicaciones. Los autores de (Musunoori et al., 2005), (Chiu et al., 2008) plantean que no definir requisitos no funcionales en fases tempranas puede producir resultados erróneos, inesperados o inaceptables. Otros autores ((Acher et al., 2008), (Wahib et al., 2008)) consideran que cada aplicación Grid posee requisitos de QoS muy específicos y que pueden variar entre usuarios que la ejecuten. El trabajo de (Kyriazis et al., 2008) considera que incluir requisitos no funcionales permite que se ejecuten aplicaciones Grid de misión crítica en entornos Grid. Solamente el trabajo de (Brandic et al., 2006) menciona que los usuarios deben poder definir la ubicación de los recursos Grid para ejecutar las tareas, esto es debido a la existencia de cuestiones legales y de seguridad que pueden impedir que la tarea se ejecute a nivel de organización o a nivel de región geográfica.

Tabla 3: Principales atributos de QoS detectados

Referencia						Otros
	Duración	Costo	Disponibilidad	Confiabilidad	Precisión	
(Musunoori et al., 2005)	X	—	—	—	X	—
(Manset et al., 2005)	—	—	—	—	—	—
(Brandic et al., 2006)	X	X	—	—	—	afinidad geográfica
(Li et al., 2007)	X	X	X	X	—	—
(Guo et al., 2007)	X	X	—	—	—	capacidad de procesamiento
(Wahib et al., 2008)	X	X	X	X	—	—

(Chiu et al., 2008)	X	-	-	-	X	ancho de banda
(Brandic et al., 2008)	X	X	X	X	-	-
(Kyriazis et al., 2008)	X	X	X	-	-	-
(Allenator y Thulasiram, 2008)	-	X	-	-	-	-
(Ranaldo y Zimeo, 2008)	X	-	-	-	-	capacidad de almacenamiento mínima, cantidad de tareas por recurso
(Acher et al., 2008)	X	X	X	X	X	reproducibilidad, previsibilidad, calidad de respuesta, prioridad

Casi la totalidad de los trabajos analizados consideran al tiempo de ejecución como aspecto de QoS que caracteriza a las aplicaciones Grid; tanto la heterogeneidad como volatilidad de los recursos Grid influirán en la duración de la ejecución como se observa en la Tabla 3. Solamente (Manset et al., 2005) y (Allenator y Thulasiram, 2008), no lo consideran. El costo o precio de ejecución de la misma forma, es considerado por la mayoría como requisito de QoS de aplicaciones Grid. La disponibilidad, confiabilidad y precisión también son atributos no funcionales importantes en el contexto del desarrollo de aplicaciones para Grid computing. Finalmente debido al caso de estudio tratado en cada trabajo, se desprenden requisitos no funcionales específicos de la aplicación: capacidad de almacenamiento mínimo, cantidad de tareas que puede ejecutar el recurso Grid, afinidad geográfica, ancho de banda, reproducibilidad, previsibilidad, calidad de la respuesta, prioridad.

Con respecto al criterio que se refiere al enfoque de ASoC utilizado en el desarrollo de aplicaciones Grid con soporte para QoS, se observa que sólo los trabajos de (Acher et al., 2008), (Musunoori et al., 2005) y (Manset et al., 2005) proponen aplicar enfoques de ASoC que permiten al usuario definir atributos de QoS desde fases tempranas de desarrollo de este tipo de aplicaciones. El resto de los trabajos no consideran este criterio de manera explícita.

Tabla 4: Herramientas propuestas para incorporar atributos de QoS

Referencia	Descripción
(Musunoori et al., 2005)	framework que incluye servicio planificador y modelos para resolver programación multi-objetivo.
(Manset et al., 2005)	framework que incluye lenguaje y editor para definir servicios y composición.
(Brandic et al., 2006)	framework que incluye editor gráfico Teuta que genera especificación par el motor de ejecución QWE.
(Li et al., 2007)	proponen sólo técnicas y métodos basados en QSC-nets que se especifica utilizando el lenguaje BPEL4WS. Sin editor.
(Guo et al., 2007)	presentan una arquitectura del sistema gestor de workflow que combina un documento BPEL generado por el editor y un documento XML conteniendo los atributos de QoS.
(Wahib et al., 2008)	framework que incluye herramientas para definir atributos de QoS a través de portlets y donde estos atributos pueden incluirse en el proceso de planificación de trabajo a través del servicio intermediario de recursos (resource broker).
(Chiu et al., 2008)	framework que soporta requisitos de QoS e incluye modelos para evaluar los costos asociados y luego compone el workflow.
(Brandic et al., 2008)	herramienta de edición y gestión de workflows Grid incluyendo

	mecanismos de meta-negociaciones y mapeos de SLA incluye el lenguaje QoWL.
(Kyriazis et al., 2008)	No se propone herramienta alguna, sólo dos tipos de servicios que permiten definir propiedades no funcionales.
(Allenor y Thulasiram, 2008)	No se propone herramienta alguna, sólo una infraestructura de políticas de precios para recursos Grid que comprende una capa de servicios Grid, que incluye el middleware, y una capa de optimización de precios situada en una capa superior.
(Ranaldo y Zimeo, 2008)	framework para incluir e integrar mediante la estrategia de matching por minimización de tiempo a los requisitos de QoS.
(Acher et al., 2008)	No se propone herramienta alguna, sino una arquitectura para aplica SPL compuesta por un framework para describir el modelo de dominio, un repositorio de servicios existentes para el dominio del problema y metamodelos.

Con respecto al último criterio de comparación, en la Tabla 4, se puede observar que todos los trabajos analizados proponen algún mecanismo (técnicas, modelos, herramientas, framework o arquitectura) para incorporar ó definir características no funcionales durante el diseño de la aplicación Grid. La propuesta más abarcativa ó integral (Brandic et al., 2006) permite que el usuario utilice un editor gráfico para diseñar y definir el workflow; se incorporan los atributos de QoS y la especificación generada por esta herramienta luego es ejecutada por un motor de workflow que lo soporta.

5 CONCLUSIONES

Se desarrolló la RSL cuyo objeto de estudio fue analizar de qué forma se incorporan y evalúan los requerimientos no funcionales o restricciones de QoS en fases tempranas del desarrollo de aplicaciones para Grid computing.

Tanto los resultados de la búsqueda de estudios primarios como la selección, extracción de datos y síntesis de datos de los trabajos seleccionados, ponen de manifiesto que el tema objeto de investigación de la RSL es de interés de la comunidad científica y que se aborda desde diferentes ópticas y estadios del proceso de desarrollo de aplicaciones Grid.

Son pocos los trabajos que proponen utilizar un enfoque de ASoC para incluir y/o evaluar los requisitos no funcionales durante todas las fase de desarrollo de aplicaciones Grid basadas en workflows de servicios.

El presente trabajo servirá como base para realizar la inclusión de atributos de QoS en fases tempranas del desarrollo de aplicaciones Grid, utilizando como enfoque de ASoC el paradigma de desarrollo de software orientado a características (Apel y Kaestner, 2009). Se diseñará un prototipo basado en este enfoque que permita incluir y evaluar los requisitos no funcionales durante todo el ciclo de vida de desarrollo de este tipo de aplicaciones para entornos Grid.

REFERENCIAS

Acher, M., Collet, P., Lahire, P., y Montagnat, J. (2008). Imaging Services on the Grid as a Product Line: Requirements y Architecture. In *SPLC (2): Software Product Lines, 12th International Conference*, pages 137–142. Lero Int. Science Centre, University of Limerick, Ireland.

Allenator, D. y Thulasiram, R. (2008). Grid resources pricing: A novel financial option based quality of service-profit quasi-static equilibrium model. In *Grid Computing, 2008 9th IEEE/ACM International Conference on*, pages 75–84.

Apel, S. y Kaestner, C. (2009). An overview of Feature-oriented software development. *Journal of Object Technology*, 8(4): 1–3.

Brandic, I., Music, D., Dustdar, S., Venugopal, S., y Buyya, R. (2008). Advanced QoS methods for Grid workflows based on meta-negotiations y SLA-mappings. *Workflows in Support of Large-Scale Science, 2008. WORKS 2008. Third Workshop on*, pages 1–10.

Brandic, I., Pillana, S., y Benkner, S. (2006). High-level composition of QoS-aware Grid workflows: An approach that considers location affinity. In *WORKS '06. Workshop on Workflows in Support of Large-Scale Science*, pages 1–10.

Brichau, J., Glyrup, M., Clarke, S., y Bergmans, L. (2002). Advanced separation of concerns. In Frohner, A., editor, *Object-Oriented Technology*, volume 2323 of *Lecture Notes in Computer Science*, pages 191–212. Springer Berlin / Heidelberg.

Caro, A., Rodríguez, A., Calero, C., Fernández-Medina, E., y Piattini, M. (2005). Análisis y revisión de la literatura en el contexto de proyectos de fin de carrera: Una propuesta. *Revista Sociedad Chilena de Ciencia de la Computación*, 6(1).

Chiu, D., Deshpande, S., Agrawal, G., y Li, R. (2008). Cost y accuracy sensitive dynamic workflow composition over grid environments. In *Grid Computing, 2008 9th IEEE/ACM International Conference on*, pages 9–16.

Eelco, V. (2010). Performing Systematic Literature Reviews with Researchr: Tool Demonstration. Technical report, The Netherlands.

Foster, I., Kesselman, C., y Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, 15(3):200–222.

Guo, L., McGough, A., Akram, A., Colling, D., Martyniak, J., y Krznaric, M. (2007). Enabling QoS for Service-Oriented Workflow on GRID. In *Computer y Information Technology, 2007. CIT 2007. 7th IEEE International Conference on*, pages 1077–1082.

Kitchenham, B. (2004). Procedures for performing systematic reviews. Technical report, UK.

Kyriazis, D., Hasselmeyer, P., Tserpes, K., Menychtas, A., y Varvarigou, T. (2008). QoS-Based Decision Services in Grids. In *Informatics, 2008. PCI '08. Panhellenic Conference on*, pages 78–82.

Li, W.-J., Liang, X.-J., Song, H.-M., y Zhou, X.-C. (2007). QoS-Driven Service Composition Modeling with Extended Hierarchical CPN. *Theoretical Aspects of Software Engineering, 2007. TASE '07. First Joint IEEE/IFIP Symposium on*, pages 483–492.

Manset, D., McClatchey, R., Oquendo, F., y Verjus, H. (2005). A Model-driven Approach for Grid Services Engineering. *arXiv.org*, cs.SE. 7 pages, 3 figures. Proc of the 18th international conference on Software y Systems Engineering y Applications. Paris November 2005.

Menascé, D. A. (2002). Qos issues in web services. *IEEE Internet Computing*, 6:72–75.

Musunoori, S., Eliassen, F., y Eide, V. (2005). QoS-driven service configuration in computational grids. *Grid Computing, 2005. The 6th IEEE/ACM International Workshop on*. RSL: seleccionado.

Ranaldo, N. y Zimeo, E. (2008). A Framework for QoS-based Resource Brokering in Grid Computing. In *Emerging Web Services Technology, Volume II*, pages 159–170.

Trejo, N. y Casas, S. (2011). Separación Avanzada de Concerns para Desarrollar Aplicaciones Grid. ICT-UNPA-26-2011, Universidad Nacional de la Patagnia Austral. Argentina. ISSN 1852-4516 -<http://168.226.35.7/secyt/ict/>

W3C Consortium (2003). QoS for Web Services: Requirements y Possible Approaches. <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>.

Wahib, M., Munawar, A., Munetomo, M., y Kiyoshi, A. (2008). SOAG: Service Oriented Architected Grids y adoption of application specific QoS attributes. In *Grid Computing, 2008 9th IEEE/ACM International Conference on*, pages 346–351. IEEE Computer Society.