Auction-based Resource Provisioning in Cloud Computing. A Taxonomy

Sara Arévalos Flor
Polytechnic School
National University of Asunción
sarevalos@pol.una.py
Paraguay

Fabio López Pires
Itaipu Technological Park
National University of Asunción
fabio.lopez@pti.org.py
Paraguay

Benjamín Barán
National University of Asunción
East National University
bbaran@pol.una.py
Paraguay

***Abstract*—Amazon Web Services marketizes its idle computing resources through its spot instances offer. These resources are offered through an auction-based scheme at extremely low prices. A running instance can be shutdown whenever the spot price rises above the user bid. Several challenges and opportunities emerge from this new computing paradigm. This work proposes for the first time a taxonomy on auction-based cloud computing resource provisioning, based on the study of the most relevant literature. The studied works are classified according to: (1) provider or user perspective, (2) problem solved, (3) optimization approach, (4) objective functions and (5) solution techniques. Finally, new prospective research subjects are identified and proposed for this promising research area.**

***Keywords*—Auction-based Resource Provisioning, Taxonomy, Cloud Computing, Cloud Broker, Spot Instances.**

***Resumen*—Amazon Web Services ofrece sus recursos computacionales ociosos por medio de instancias puntuales (*spot instances*). La oferta se realiza a través de un esquema de subasta a precios extremadamente bajos. Las instancias puntuales en ejecución pueden ser apagadas cuando el precio puntual supera la oferta del cliente. Varios desafíos y oportunidades emergen de este nuevo paradigma computacional. Este trabajo propone por primera vez una taxonomía basada en la literatura más relevante considerando esquemas de provisión de recursos en Computación en la nube *(Cloud Computing)* basados en subastas. Los trabajos estudiados se clasifican según: (1) la perspectiva del proveedor o del usuario, (2) el problema resuelto, (3) el enfoque de optimización, (4) las funciones objetivo y (5) las técnicas de solución. Finalmente, se identifican y proponen nuevos posibles trabajos de investigación para esta prometedora área de investigación.**

***Palabras Clave*—Provisión de Recursos Basados en Subasta, Taxonomía, Cloud Computing, Intermediario, Instancias Puntuales.**

I. Introducción

El reciente avance en Computación en la Nube, en adelante *Cloud Computing*, ha introducido nuevos modelos de negocio basados en el concepto de la computación como una utilidad, en forma similar a como hoy se adquiere la electricidad o el agua [1]. Los proveedores de servicios de *Cloud Computing* enfocados en el modelo de Infraestructura como Servicio (*Infrastructure as a Service, IaaS*) ofrecen recursos computacionales que pueden ser alquilados de acuerdo a las necesidades de los usuarios [2], con métodos de pago basados en el uso de estos recursos y con diversos esquemas de precio. En este contexto, Amazon Elastic Compute Cloud (EC2) es considerado el principal proveedor público de IaaS [3], por lo que este trabajo se basará en los esquemas de precio utilizados por Amazon EC2, principalmente el esquema de precio basado en subasta.

Amazon EC2 categoriza las máquinas virtuales (instancias) que ofrece (según las prestaciones de recursos con que cuenta) en 32 tipos de instancias [4]. Los diferentes tipos de instancias pueden ser utilizados bajo tres posibles esquemas: (1) bajo demanda (*on demand instances*), (2) reservadas (*reserved instances*) o (3) puntuales (*spot instances*) [4]. El precio se determina en función al tipo de instancia contratada y al esquema de utilización.

Las instancias en el esquema bajo demanda son contratadas considerando precios fijos. El usuario paga un monto dado por cada hora de uso de un determinado tipo de instancia. Es importante destacar que la contratación de una instancia bajo demanda no implica un compromiso a largo plazo del usuario con el proveedor.

Por otro lado, el esquema de instancias reservadas está destinado a usuarios con necesidades de largo plazo. Las instancias reservadas requieren un pago por adelantado a fin de garantizar la capacidad de recursos reservada. La ventaja de este esquema en comparación al esquema de instancias bajo demanda es obtener precios considerablemente menores [4].

En el esquema de instancias puntuales (*spot instances*), el precio de los recursos es fijado dinámicamente por los proveedores de servicios según la oferta y la demanda existentes en un proceso similar al de una subasta. El precio de las instancias puntuales es denominado precio puntual. Los usuarios que desean utilizar instancias puntuales deben especificar el precio máximo que están dispuestos a pagar por cada hora de ejecución de un determinado tipo de instancia. Cuando el precio puntual es menor al precio ofertado por un usuario, éste obtiene los recursos computacionales solicitados. Si el precio puntual sube por encima del precio ofertado por el usuario, Amazon EC2 termina la instancia [4].

Una explicación más detallada del funcionamiento del esquema de instancias puntuales de Amazon EC2 se presenta en la Sección II.

978-1-4673-9143-6/15/$31.00

Lo novedoso de las instancias puntuales es que a través de ellas, Amazon EC2 comercializa recursos computacionales que de otra manera estarían ociosos, mejorando así la previsibilidad de la carga en sus Centros de Datos y logrando un uso eficiente de sus recursos computacionales. Amazon EC2 fija el precio puntual en función a la oferta (recursos ociosos) y la demanda (solicitudes de instancias puntuales) existentes. Según [5], en un Centro de Datos tradicional, donde todavía no se utilizan esquemas de virtualización, la mayoría de los servidores funcionan a una tasa de utilización baja, la cual se estima entre un 6 y un 12 %. En un proveedor de IaaS que utiliza solamente esquemas similares a las instancias bajo demanda y/o reservadas, todavía existe un considerable porcentaje ocioso de capacidad de cómputo. Las instancias puntuales permiten así que Amazon EC2 utilice sus recursos de una manera eficiente [6], utilizando así la mayor cantidad posible de recursos disponibles en pos de una mejor rentabilidad de la empresa.

Para los usuarios, el bajo precio constituye un gran incentivo para utilizar el esquema de instancias puntuales. Sin embargo, no todos los usuarios pueden hacer frente a la imprevisibilidad y poca confiabilidad de los recursos computacionales obtenidos bajo esquemas de instancias puntuales. No obstante, las instancias puntuales siguen resultando bastante atractivas para usuarios con aplicaciones tolerantes a fallos, paralelas o distribuidas tales como: procesos por lote, simulaciones científicas, procesamiento de imágenes, codificación de video y análisis de datos [4].

Los esquemas de precios basados en subasta como el de Amazon EC2 presentan grandes desafíos y oportunidades para los proveedores de IaaS que poseen recursos ociosos en su infraestructura. Para los usuarios que buscan explotar las ventajas económicas de este esquema de precios, es necesario encontrar mecanismos que mitiguen las desventajas que presentan. Actualmente existen empresas que actúan de intermediario entre las necesidades de los usuarios finales, en adelante clientes y la oferta de los proveedores. Para estos intermediarios, las instancias puntuales presentan una clara oportunidad de aumentar sus ingresos si encuentran la manera de aprovechar estos recursos computacionales de forma confiable.

Debido al gran interés que existe actualmente sobre los esquemas de precios basados en subasta, numerosos trabajos de investigación han estudiado diferentes problemas relacionados a la gestión de infraestructuras virtuales en ambientes de *Cloud Computing* con este esquema. Sin embargo, no existen trabajos de investigación publicados que presenten el estudio general de los principales tópicos de investigación relacionados a los esquemas de precio basados en subasta para ambientes de *Cloud Computing*, específicamente para IaaS. Consecuentemente, este trabajo presenta una revisión sistemática de la literatura relacionada y propone por primera vez una taxonomía donde los artículos estudiados son clasificados según: (1) perspectiva (proveedor o intermediario), (2) problema estudiado, (3) enfoque de optimización, (4) funciones objetivo y (5) técnicas de solución.

El resto de este trabajo está organizado de la siguiente manera: en la Sección II se definen los principales conceptos y terminologías utilizados en este trabajo; en la Sección III se describe la taxonomía propuesta; mientras que las consideraciones necesarias para la resolución de los problemas tratados en la taxonomía son detallados en la Sección IV y, finalmente, las conclusiones y trabajos futuros son presentados en la Sección V.

II. Conceptos y terminología

Cloud Computing es un área de investigación reciente y aún existen ambiguedades en la terminología utilizada en la literatura. Por otro lado, los esquemas de precios basados en subasta para mercados de *Cloud Computing* representan un área de investigación aún reciente por lo que muchos trabajos utilizan diversos términos para referirse específicamente a lo mismo. En consecuencia, en esta sección se definen algunos conceptos y términos a ser utilizados en este trabajo.

A. Definiciones

- Intermediario (*Broker*): Entidad que actúa de intermediario entre los clientes finales y los proveedores de servicios. El intermediario obtiene recursos computacionales de los proveedores buscando minimizar sus costos. Provee a los clientes un servicio integrado para la administración, monitoreo y asignación de recursos [6], minimizando la complejidad de utilizar recursos computacionales en la nube desde la perspectiva de un simple usuario final.
- Infraestructuras intermitentes: Como se ha definido anteriormente, una instancia puntual tiene una duración desconocida. El proveedor de IaaS puede apagar la instancia puntual si el precio puntual es mayor que el precio ofertado por el usuario. En consecuencia, una infraestructura intermitente es un conjunto de recursos computacionales compuesto por instancias puntuales.
- Mecanismos de tolerancia a fallos: Son técnicas que permiten minimizar el impacto del apagado de una instancia puntual por parte del proveedor [7]. Por ejemplo, un trabajo que será interrumpido por el apagado de la instancia en la que se está ejecutando puede guardar su estado de tal manera a continuar en otra instancia en el futuro (*checkpointing*) [8] [9]. Cabe destacar que el problema de determinar el momento más adecuado para llevar a cabo el *checkpointing* ha sido mitigado desde el momento en que Amazon EC2 informa a los clientes el momento de apagado de sus instancias puntuales de manera anticipada [10].
- Tiempo total de proceso (*Makespan*): Cuando existen varias tareas, las mismas pueden ser ejecutadas con diferentes tiempos de finalización dependiendo del conjunto de instancias al cual son asignadas y la secuencia de ejecución. El tiempo total de proceso, es el tiempo que demanda la ejecución de todas las tareas [11].
- Confiabilidad (*Reliability*): En el contexto del presente trabajo, la confiabilidad es el grado de certidumbre de que una tarea se terminará de procesar en un periodo de tiempo predeterminado. La confiabilidad se consigue mediante la aplicación de los mecanismos de tolerancia a fallos [12], utilizando alguna política que maximice la previsibilidad de terminar un trabajo para el momento previsto.

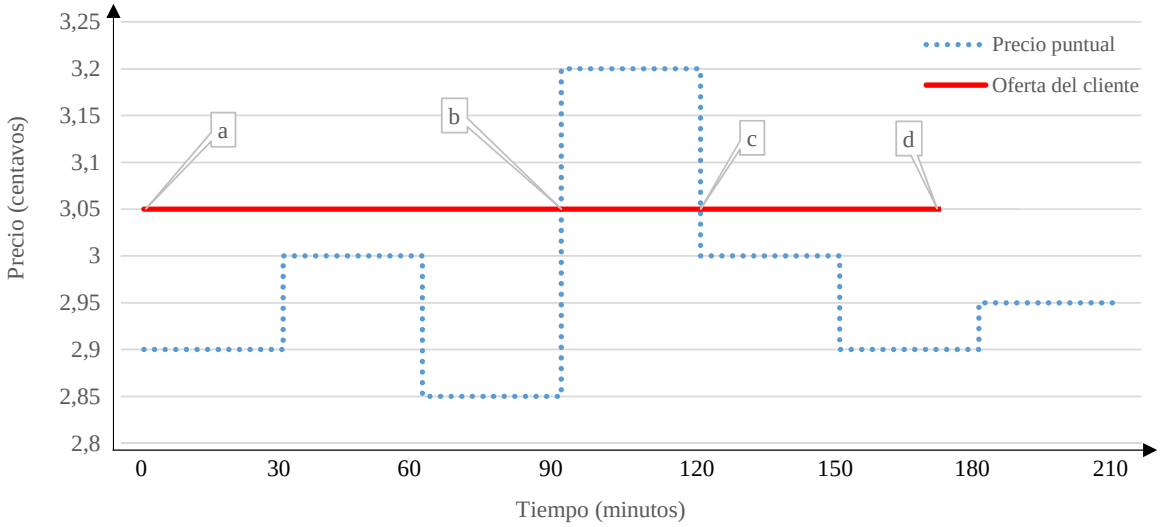

Figura 1: Esquema de instancias puntuales de Amazon EC2.

B. Funcionamiento del esquema de subasta de instancias puntuales de Amazon EC2

En la Figura 1 se presenta un ejemplo de funcionamiento del esquema de subasta de instancias puntuales de Amazon EC2.

En primer lugar, el cliente fija el valor máximo que está dispuesto a pagar cada hora por un tipo de instancia. En este ejemplo, el cliente oferta 3,05 centavos de US$ (Figura 1, Paso a). Como la oferta del cliente está por encima del precio puntual, Amazon EC2 provee la instancia solicitada de manera inmediata. De acuerdo a los términos de uso de Amazon EC2, el cliente debe abonar por cada hora el valor del precio puntual al momento de iniciarse su instancia.

Amazon EC2 apaga la instancia cuando el precio puntual supera el precio máximo fijado por el cliente (Figura 1, Paso b). Si la instancia es apagada por Amazon EC2, el cliente no paga por la porción de hora utilizada. En el ejemplo, el cliente paga por la primera hora y no paga por la mitad utilizada de la segunda hora.

Una oferta por una instancia puntual puede mantenerse hasta que el precio puntual baje. Cuando el precio puntual vuelve a estar por debajo de una oferta persistente, una nueva instancia del tipo solicitado es proveída (Figura 1, Paso c). Cuando es el cliente quien detiene su instancia antes de cumplirse una hora entera, debe pagar por la porción de hora utilizada (Figura 1, Paso d).

Una novedad introducida recientemente al esquema de instancias puntuales de Amazon EC2 es que los clientes son notificados dos minutos antes que sus instancias serán apagadas [10]. Este tiempo de aviso permite al usuario ejecutar algún mecanismos de tolerancia a fallos. En la Sección IV se presentarán algunos de los mecanismos de tolerancia a fallos estudiados en la literatura.

III. Taxonomía Propuesta

Para la elaboración de la taxonomía propuesta se buscaron trabajos relacionados a instancias puntuales publicados en los últimos 4 años en reconocidas bibliotecas científicas como IEEE Xplore, ACM Digital Library, Elsevier y Springer.

De las publicaciones encontradas, los autores del presente trabajo han seleccionado aquellas que consideran más representativas del estado del arte. Con la literatura seleccionada, se ha realizado un estudio más profundo, lo que llevó a proponer la clasificación de trabajos considerando los siguientes criterios:

En el primer nivel, los trabajos relacionados a instancias puntuales pueden ser analizados desde la perspectiva del proveedor o desde la perspectiva del usuario (intermediario o cliente).

Una vez definida la perspectiva desde donde se analiza el problema en cuestión, los trabajos pueden ser clasificados según el problema que estudian.

El tercel nivel corresponde al enfoque de optimización utilizado en las funciones objetivo, clasificándose en mono-objetivo o multi-objetivo.

Los objetivos específicos a ser optimizados en la formulación del problema constituyen el cuarto nivel de clasificación. Se pueden tener objetivos de minimización y/o maximización.

El último criterio de clasificación lo constituyen las técnicas utilizadas para resolver el problema, pudiendo ser éstas: técnicas estadísticas, heurísticas, meta-heurísticas o nuevos algoritmos.

En la Figura 2 se presenta la taxonomía propuesta según los criterios mencionados anteriormente para problemas de provisión de recursos computacionales en *Cloud Computing* basados en esquemas de subasta. A continuación se detalla cada uno de los niveles de clasificación de la taxonomía propuesta.

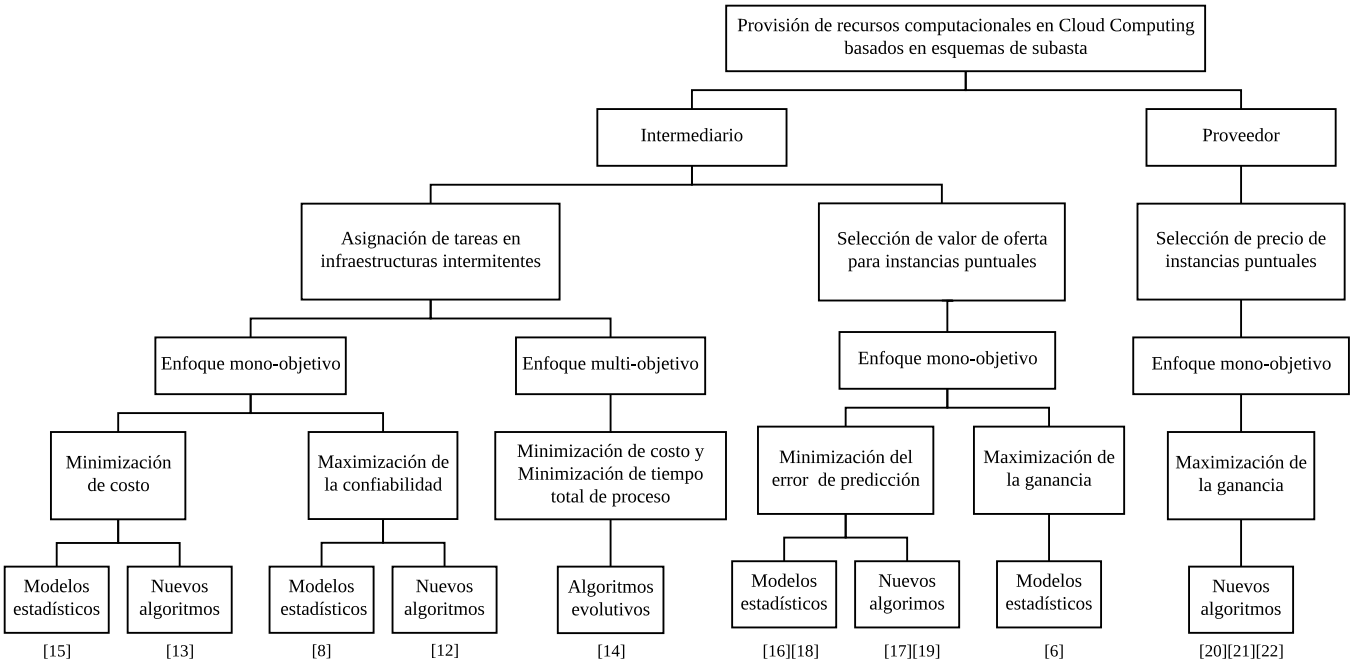

Figura 2: Taxonomía de la provisión de recursos computacionales en *Cloud Computing* basados en esquemas de subasta.

A. Perspectiva (proveedor o intermediario)

En la presente taxonomía, los problemas que debe resolver un cliente de IaaS son un subconjunto de los problemas del intermediario. Por lo tanto, a fin de simplificar el modelo, se estudian los trabajos desde las perspectivas del proveedor y desde la perspectiva del intermediario. En la Figura 3 se observa la interacción entre los diferentes actores del mercado de infraestructuras intermitentes.

1) Perspectiva del proveedor: El proveedor busca administrar y aprovechar de manera óptima sus recursos computacionales. Bajo esta premisa, los recursos ociosos pueden ser ofertados a través de un esquema de precios basado en subastas. Mediante la aplicación del esquema de instancias puntuales, el proveedor aprovecha de manera eficiente sus recursos computacionales ociosos, por ejemplo, maximizando sus ganancias.

2) Perspectiva del intermediario: Para los clientes finales, el esquema de instancias puntuales es atractivo por el bajo precio. Sin embargo, elegir un precio de oferta y minimizar el impacto del apagado imprevisto de las instancias puntuales son complejidades que el cliente debe enfrentar para dar un uso eficiente a los recursos contratados.

Un intermediario puede abstraer a los clientes de estas complejidades y obtener un beneficio económico al encargarse de la gestión de las tareas necesarias (como por ejemplo admisión de tareas, ejecución y monitoreo) y la gestión del conjunto de instancias (ofertas, selección, creación y terminación de instancias) [13]. El intermediario entonces, negocia directamente las contratación de servicios entre los clientes y los proveedores de IaaS [2], recibiendo un beneficio económico por esta intermediación.

Para el cumplimiento de sus objetivos, el intermediario obtiene toda la información disponible sobre una tarea, acuerda un nivel de calidad de servicio (SLA) con el cliente y utiliza esta información para tomar decisiones de provisionamiento de instancias y calendarización de tareas, buscando maximizar sus beneficios económicos sin desatender las necesidades de sus clientes.

B. Problemas Estudiados

La taxonomía propuesta en este trabajo identifica tres problemas principales ya estudiados en la literatura especializada. Desde la perspectiva del intermediario se consideran los siguientes problemas: (1) la asignación de tareas en infraestructuras intermitentes y (2) la selección del valor de la oferta para instancias puntuales. A su vez, desde la perspectiva del proveedor se estudian (3) las técnicas para la selección del precio de las instancias puntuales. Cada uno de estos tres problemas es explicado en detalle a continuación:

Problema 1. Asignación de tareas en infraestructuras intermitentes: Este problema consiste en determinar la mejor manera de ejecutar tareas computacionalmente intensivas en un conjunto de máquinas virtuales compuestas únicamente por instancias puntuales, las cuales pueden ser de diferentes tipos (e.g. diferentes capacidades de cómputo).

El intermediario administra la ejecución de las tareas, para lo cual se encarga de recibir las solicitudes por parte de sus clientes y obtener un conjunto de instancias puntuales donde ejecutarlas.

Finalmente, el intermediario debe calendarizar y asignar cada tarea según la disponibilidad de sus instancias. Según [13], los pasos para la asignación de las tareas son los siguientes:

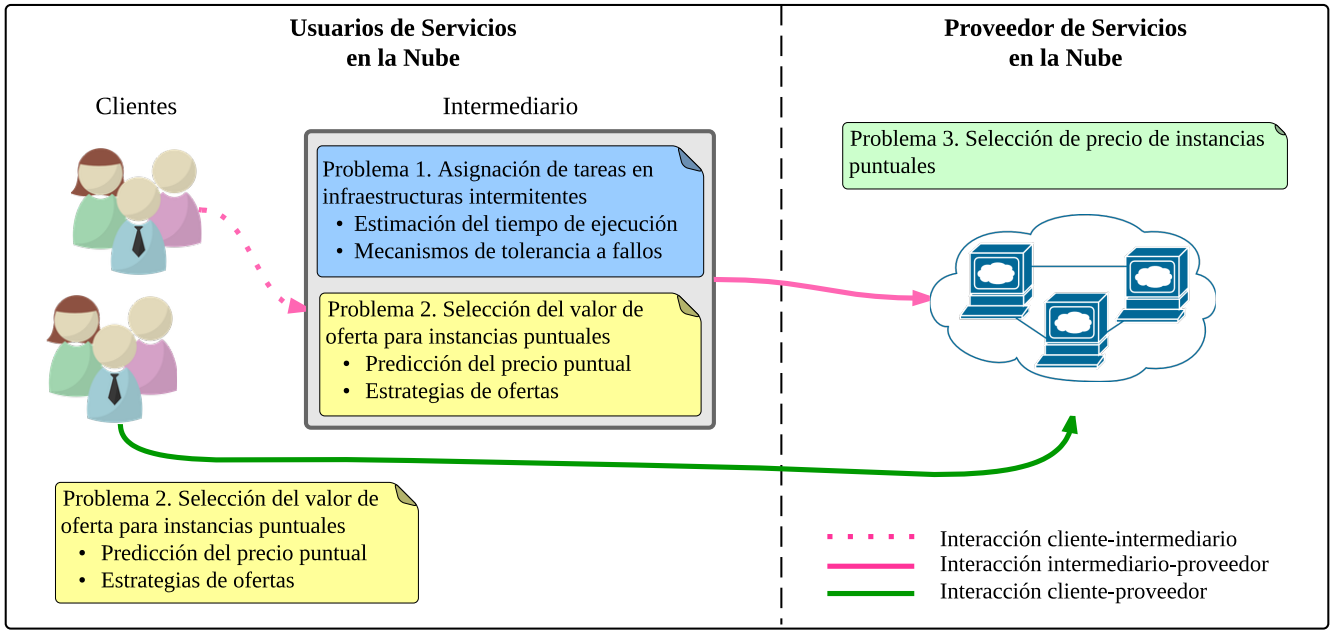

Figura 3: Interacción entre los principales actores de Computación en la Nube y los problemas considerados en la taxonomía propuesta

Paso 1: Cuando una tarea es recibida para su ejecución, ésta es agregada a una lista de tareas a ser procesadas.
Paso 2: A intervalos regulares de tiempo, un método de predicción es utilizado para estimar el tiempo aproximado de ejecución de la tarea en cada tipo de instancia disponible.
Paso 3: El intermediario intenta ubicar la tarea en una instancia ociosa con tiempo suficiente antes que finalice una hora completa.
Paso 4: Si no tiene éxito, intenta ubicar la tarea en una instancia que actualmente tiene tareas activas pero que se espera estará ociosa al cabo de poco tiempo. En este paso, es necesaria la estimación de tiempo de todas las tareas activas en las instancias, además de la tareas entrantes.
Paso 5: Si la tarea todavía no puede ser ubicada, se decide si es ventajoso extender el tiempo de ejecución de una instancia, contratar una nueva instancia o posponer la decisión de ubicación de la tarea. Estas decisiones se realizan de acuerdo al factor de urgencia de la tarea y las condiciones del precio puntual del momento.

El problema de asignación de tareas en infraestructuras intermitentes es estudiado en [8], buscando minimizar el costo monetario mediante la aplicación de un mecanismo de tolerancia a fallos denominado por los autores como *checkpoint* adaptativo.

En [12], Voorsluys y Buyya proponen un algoritmo para resolver el problema de asignación de tareas a través de estrategias de ofertas y técnicas de tolerancia a fallos.

Voorsluys et al. aplican y comparan varias técnicas de estimación de tiempo de ejecución de tareas. El tiempo de ejecución es utilizado para determinar cuál es el mejor conjunto de instancias y el mejor momento para llevar a cabo la ejecución de cada tarea [13].

En el trabajo [14], Vintila et al. estudian el problema enfocándose en la minimización del tiempo total de proceso (*makespan*). En este trabajo se considera que el conjunto de máquinas virtuales está compuesto por una combinación de instancias puntuales e instancias bajo demanda.

Por último, Tang et al. proponen construir una matriz de probabilidad de precios a fin de establecer el momento más oportuno para ejecutar una tarea utilizando el tiempo de ejecución como restricción. El objetivo es ejecutar las tareas cuando se espera que el precio sea menor y puedan ser finalizadas dentro de la restricción de tiempo [15].

Las consideraciones relacionadas al problema de asignación de tareas en infraestructuras intermitentes, tales como las técnicas de estimación del tiempo de ejecución de tareas y las técnicas de tolerancia a fallos son explicadas en la sección IV.

Problema 2. Selección del valor de oferta: Un intermediario debe obtener las instancias necesarias para satisfacer las necesidades de sus clientes a un precio que le permita maximizar sus ganancias. Para esto, el intermediario necesita un mecanismo para determinar cuánto debe ofertar por cada tipo de instancia y el momento en que debe realizar la oferta.

En [6], Song et al. proponen un algoritmo que elige el monto de la oferta en base al precio puntual actual a fin de maximizar el beneficio económico del intermediario.

Tang et al. proponen inicialmente una estrategia de ofertar siempre por el precio más alto, lo cual garantiza que todas las tareas serán ejecutadas en un tiempo mínimo. Según los registros históricos de precios puntuales el precio máximo es cuando mucho 1.1 veces mayor al mínimo, por lo tanto con esta estrategia sólo se pagará hasta 10 % más que el precio puntual mínimo [15].

A pesar de que no existen desventajas aparentes en ofertar siempre por el precio más alto, existen incentivos para seleccionar estrategias donde se oferten valores muy cercanos al precio puntual actual. En un escenario donde la mayoría de los usuarios ofertan valores altos, los proveedores incrementarán el precio puntual para maximizar sus ganancias [12].

En [9], Yi et al. destacan la oportunidad de aprovechar el hecho que Amazon EC2 no cobra por la fracción de hora previa al apagado de la instancia puntual en una situación de incremento del precio puntual. Bajo esta premisa, un cliente podría ofertar un valor muy próximo al precio puntual actual con la esperanza de que ocurra un fallo antes de cumplirse una hora, evitando de esta manera pagar por la fracción utilizada.

Otros trabajos basados en la predicción del precio puntual también fueron incluídos bajo el problema de selección del valor de oferta. Según [16], la predicción del precio puntual se ha convertido en una de las principales áreas de la investigación moderna en *Cloud Computing*.

En el trabajo [17], Ben-Yehuda et al. realizaron ingeniería inversa sobre el mecanismo de fijación de precios de Amazon EC2. Consideran que los precios de las instancias puntuales no están determinados por las condiciones del mercado sino que por el contrario son generados de manera aleatoria dentro de un estrecho margen de precios. En base a esto, construyen un modelo que genera precios consistentes con los precios históricos reales.

En [16] y [18] se propone un modelo estadístico que modela la dinámica del precio puntual a partir del análisis del histórico de precios puntuales de Amazon EC2. A su vez, Kumar y Dutta, presentan un modelo de predicción de precios desde la perspectiva del intermediario [19].

Las estrategias para determinar el precio a ofertar por instancias puntuales son detalladas en la Sección IV.

Problema 3. Selección de precio de instancias puntuales: Desde la perspectiva del proveedor, el esquema de fijación del precio puntual debe ser cuidadosamente diseñado y los impactos presentes y futuros deben ser considerados.

En el presente, el proveedor puede fijar un precio puntual mayor para aumentar sus ingresos. El precio puntual mayor desplaza peticiones con ofertas menores a un tiempo posterior, reduciendo los ingresos futuros. También debe ser considerada la calidad de servicio, ya que una espera muy prolongada para obtener los recursos solicitados puede ocasionar la pérdida de potenciales clientes [20].

En [21] y [22] los autores proponen fijar los precios puntuales de manera dinámica con el objetivo de maximizar los ingresos del proveedor.

Por otro lado, en [20], Wang et al. modelan el impacto del precio puntual en los ingresos presentes y futuros del proveedor. Para esto, proponen un modelo que tiene en cuenta el retardo en la atención de las peticiones de los clientes para fijar el precio puntual.

En la Figura 3 se detalla la relación de los problemas estudiados en un esquema de precios basado en subasta, según las perspectivas de los actores involucrados.

C. Enfoque de Optimización

Dependiendo del número de objetivos considerados, la formulación de los problemas descritos en la sección III.B pueden ser clasificados bajo uno de los siguientes enfoques:

1) Enfoque Mono-Objetivo: El enfoque mono-objetivo considera la optimización de solo una función objetivo o la optimización individual de más de un objetivo, uno a la vez. En la literatura estudiada, la mayoría de los trabajos optimiza un único objetivo, tal como puede observarse en la Tabla I.

2) Enfoque Multi-Objetivo: Un Problema de Optimización Multiobjetivo *(Multi-objective Optimization Problem - MOP)* consiste en un conjunto de p variables de decisión, un conjunto de q funciones objetivos y un conjunto de r restricciones. Las funciones objetivos y las restricciones son funciones de las variables de decisión. En [23] se formaliza un MOP como: Optimizar:

$$y = f(x) = (f_1(x), f_2(x), \ldots, f_q(x)) \tag{1}$$

sujeto a

$$e(x) = (e_1(x), e_2(x), \ldots, e_r(x)) \geq 0 \tag{2}$$

donde $x = (x_1, x_2, \ldots, x_p) \in X \subset \Re^p$ es un vector de decisión, X denota el espacio de decisión de $f(x), y = (y_1, y_2, \ldots, y_q) \in Y \subset \Re^q$ es un vector objetivo mientras que Y denota el espacio objetivo de $f(x)$. Un *Conjunto de Soluciones Factibles* $\Omega \in X$ es definido como un conjunto de vectores de decisión que satisfacen las restricciones dadas en (2). Sean dos soluciones $u, v \in \Omega$, se dice que u domina a v (denotado como $u \succ v$) si u es mejor o igual que v en cada función objetivo y estrictamente mejor en al menos un objetivo. Así, se define el *Conjunto Pareto Óptimo* como $P^* = \{u \in \Omega | \nexists v \in \Omega \text{ tal que } v \succ u\}$ mientras que el espacio objetivo de P^* es conocido como *Frente Pareto Óptimo*, denotado como $F^* = f(P^*)$.

En los trabajos estudiados se resuelven diferentes tipos de problemas principalmente mediante un enfoque mono-objetivo. De la literatura estudiada, solamente el trabajo de Vintila et al. posee un enfoque multi-objetivo [14], donde se utiliza dominancia Pareto para comparar las soluciones que están en relación de compromiso entre sí. En dicho trabajo, se busca simultáneamente la minimización del tiempo total de proceso y la minimización del costo de las instancias puntuales para el intermediario.

De acuerdo a las funciones objetivos identificadas y la naturaleza de los problemas estudiados, la optimización multi-objetivo permitiría realizar formulaciones más completas y realistas, dado que en la práctica son varios los objetivos que deben ser considerados al momento de tomar una decisión. Esta decisión generalmente es una alternativa de compromiso entre los diferentes factores considerados.

Según la Tabla I, no existen enfoques en un contexto multi-objetivo para la resolución de los problemas de selección del valor de oferta y de selección del precio de instancias puntuales, lo cual representa un área potencial de investigación, lo que se denota en dicha tabla como *OI** (Oportunidad de Investigación).

Tabla I: Provisión de recursos computacionales en Cloud Computing basados en esquemas de subasta. Los elementos de la tabla representan los artículos en el universo estudiado, indicándose las oportunidades de investigación (**OI***) para los casos en que no se encontró ningún artículo.

Perspectiva	**Problema**	**Enfoque**	**Objetivos**	**Técnicas de solución**		
				Modelos estadísticos	*Nuevos algoritmos*	*Algoritmos evolutivos*
Intermediario	*Asignación de tareas en infraestructuras intermitentes*	*Mono Objetivo*	1.Minimización del costo	[15]	[13]	**OI***
			2.Maximización de la confiabilidad	[8]	[12]	**OI***
		Multi Objetivo	3.Minimización del tiempo total de proceso y 1.Minimización del costo	**OI***	**OI***	[14]
	Selección del valor de oferta para instancias puntuales	*Mono Objetivo*	4.Minimización del error de predicción	[16] [18]	[17] [19]	**OI***
			5.Maximización de las ganancias	[6]	**OI***	**OI***
		Multi Objetivo	**OI***	**OI***	**OI***	**OI***
Proveedor	*Selección de precio de instancias puntuales*	*Mono Objetivo*	5.Maximización de las ganancias	**OI***	[20] [21] [22]	**OI***
		Multi Objetivo	**OI***	**OI***	**OI***	**OI***

D. Funciones Objetivo

Los problemas identificados en la presente taxonomía son estudiados en la literatura en base a diferentes formulaciones y diversos objetivos de optimización. En la Tabla I se presentan los cinco grupos considerados de funciones objetivo y los trabajos que abordan cada grupo. A continuación se describe cada uno de estos grupos de funciones objetivo.

1) Minimización del costo: El intermediario necesita obtener los recursos computacionales al menor costo posible a fin de maximizar sus ganancias. Para ello, el intermediario debe identificar el conjunto de instancias puntuales que permita que las tareas se ejecuten a un precio bajo respetando el nivel de confiabilidad esperado.

En [13], Voorsluys et al. presentan una política de asignación de recursos que permite ejecutar un conjunto de tareas de manera económica. Para decidir cuáles son los mejores tipos de instancias para cada tarea, la política de asignación depende de la estimación del tiempo de ejecución de las tareas.

Una estrategia de oferta óptima de precios puntuales es presentada en [15] por Tang et al.

2) Maximización de la confiabilidad: Considerando la reducida confiabilidad de las instancias puntuales, varios trabajos se enfocan en mejorarla. El aumento de la confiabilidad se logra principalmente a través de mecanismos de tolerancia a fallos; tales como los mencionados en la Sección IV.

Cabe resaltar que, si el apagado de una instancia puntual ocasiona una violación del acuerdo de nivel de servicio entre un intermediario y su cliente, el intermediario podría tener que incurrir en un costo por dicha violación [24].

Jangjaimon y Tzeng presentan un modelo que aumenta la confiabilidad al utilizar recursos puntuales [8]. La maximización de la confiabilidad se obtiene al predecir de manera precisa los momentos en los que debe aplicarse una determinada técnica de tolerancia a fallos *(checkpointing)*.

En [12], Voorsluys y Buyya proponen una política de asignación de un conjunto de tareas en una infraestructura intermitente. El esquema considera estrategias de estimación de precio y la aplicación de tres técnicas de tolerancia a fallos: *checkpointing, task duplication y task migration*. Las tres técnicas son explicadas en la Sección IV.

3) Minimización del tiempo total de proceso (makespan): La optimización del tiempo total de proceso es importante porque algunas de las tareas pueden tener un tiempo límite de ejecución impuesto por los clientes. Para el intermediario es conveniente ejecutar las tareas en el menor tiempo posible para liberar recursos y atender nuevas peticiones.

En [14], Vintila et al. proponen un algoritmo que optimiza este objetivo al mismo tiempo que minimiza el costo. El algoritmo permite al usuario favorecer el procesamiento rápido de las tareas a un bajo costo monetario.

4) Minimización del error de predicción: Para los trabajos que proponen mecanismos de predicción del precio puntual, la presente taxonomía considera que su función objetivo es la minimización del error de predicción.

Existe una discusión sobre qué factores influyen en el precio de las instancias puntuales. Así, en [17] y [21] se afirma que es improbable que dicho precio sea fijado de acuerdo a la demanda del mercado. Los autores creen que existe un factor artificial que se incluye en el cálculo del precio puntual, en contradicción a lo afirmado por Amazon EC2 en la descripción oficial de sus servicios [4].

En [16], Javadi et al. proponen un modelo estadístico para predecir el precio de las instancias puntuales de Amazon EC2. El modelo estadístico se construye en base al historial de precios de las instancias puntuales y mediciones de los intervalos de tiempo entre cambios.

En [17], Ben-Yehuda et al. realizan ingeniería reversa para determinar el mecanismo de fijación de precios puntuales de Amazon EC2. Se estudia el histórico de precios puntuales y se construye un modelo que genera resultados consistentes comparados con los precios puntuales reales.

Por último, Kumar y Dutta proponen un algoritmo que también realiza una predicción del precio puntual de Amazon EC2 [19]. Los autores asumen que el precio puntual está influenciado por una tendencia global y un patrón local que ocurre cuando existe un cambio repentino del precio.

El error de predicción puede ser medido según varias métricas, las cuales serán desarrolladas en la Sección IV.

5) Maximización de las ganancias: La maximización de las ganancias es un objetivo común tanto para el intermediario como para el proveedor. Ambos buscan aumentar sus ingresos aprovechando la creciente demanda por utilizar los servicios de *Cloud Computing*. No obstante, cada uno implementa una estrategia diferente de acuerdo a su modelo de negocio.

El modelo de negocio del intermediario consiste en aceptar tareas de los clientes y buscar instancias a menor precio en los proveedores a fin de maximizar su beneficio.

En [6], Song et al. presentan el problema de diseñar una estrategia para elegir el precio de oferta para una instancia puntual desde la perspectiva de un intermediario de servicios de *Cloud Computing*.

Desde la perspectiva del proveedor, la selección del precio para la instancia puntual no es una tarea trivial. El problema se basa en determinar cual es el mejor precio que el proveedor debe asignar a los recursos ociosos de manera a maximizar sus ganancias.

Wang et al. presentan en [20] un modelo del impacto del precio de las instancias puntuales en los ingresos del presente y del futuro. El problema de maximización de ganancias se formula como un problema de optimización promedio en el tiempo.

En [22], Zaman y Grosu proponen un mecanismo para determinar el precio que deben pagar los clientes atendidos y el conjunto de instancias que deben ser creadas. El mecanismo asegura que se utilice la máxima cantidad de recursos disponibles y que ninguna instancia puntual sea asignada a un precio que no implique mayores ganancias para el proveedor.

En [21], Xu y Li adoptan una plataforma de administración de ingresos del campo de la economía. El problema de maximización de las ganancias se formula con precios dinámicos y considerando un programa estocástico dinámico.

E. Técnicas de solución

Las principales técnicas de solución utilizadas en el estado del arte son los modelos estadísticos y por otro lado los algoritmos específicos propuestos por los autores. En un caso se propone un enfoque basado en algoritmos evolutivos. En la Tabla I se resumen las técnicas de solución utilizadas para la resolución de los problemas estudiados, identificando en cada caso las oportunidades de investigación (OI) existentes.

1) Modelos estadísticos: Algunos trabajos enfocados en resolver el problema 2 (selección del valor de oferta) utilizan técnicas estadísticas para su solución. En [16] y [18] se realiza un análisis de precios del histórico de precios puntuales de Amazon EC2. A partir de este estudio, se propone un modelo estadístico basado en una distribución Gaussiana.

Los trabajos [6] y [15] proponen técnicas de solución basadas en la utilización de cadenas de Markov. A su vez en [8] se propone realizar *checkpointing* adaptativo e incremental a fin de reducir costos. En este trabajo la predicción del momento oportuno para la aplicación de la técnica de tolerancia a fallos mencionada está basada también en modelos de cadenas de Markov.

En la Tabla I se observa que varios trabajos han propuesto un modelo estadístico para la resolución de diferentes problemas. Sin embargo, no se observan propuestas que utilicen esta técnica para la resolución del problema 3 (selección de precio de instancias puntuales), por lo que esto representa una clara oportunidad de investigación (OI). Por otro lado, un escenario con enfoque multi-objetivo tampoco es estudiado con modelos estadísticos, constituyendo una potencial área de investigación.

2) Algoritmos evolutivos: De los trabajos estudiados, [14] es el único que propone un algoritmo evolutivo para la minimización del tiempo total de proceso y el costo monetario, sin embargo, según se observa en la Tabla I, existen oportunidades de investigación para aplicar algoritmos evolutivos en la optimización de los demás objetivos estudiados.

Adicionalmente, otras meta-heurísticas podrían ser consideradas, como por ejemplo: *Ant Colony Optimization* (ACO) [25], *Particle Swarm Optimization* (PSO) [26], *Simulated Annealing* (SA) [27], *Harmony Search* (HS) [28] por citar algunas de las más relevantes.

3) Nuevos algoritmos: Bajo esta clasificación se consideran todas las demás técnicas de solución encontradas que no caben dentro de las categorías previamente citadas. Dentro de la literatura estudiada, se considera que los siguientes trabajos utilizan nuevas propuestas de algoritmos:

En [12], Voorsluys y Buyya proponen un algoritmo basado en una política de asignación de un conjunto de tareas utilizando estrategias de estimación de precio y técnicas de tolerancia a fallos.

Por su parte en [13], se presenta un algoritmo que propone la asignación de tareas en infraestructuras intermitentes utilizando diferentes métodos de estimación del tiempo de ejecución de las tareas.

En [17], Ben-Yehuda et al. utilizan un algoritmo regresivo para demostrar que el precio puntual de Amazon EC2 no sólo está basado en la oferta y la demanda sino que también existe una función artificial que afecta al precio.

En el trabajo [19] se propone una técnica de predicción del precio puntual. Para la fase de entrenamiento de las variables del esquema propuesto, Kumar y Dutta utilizaron el algoritmo del gradiente descendiente y optimización lineal de mínimos cuadrados.

En [20], Wang et al. implementan un algoritmo en línea que utiliza la técnica de optimización de *Lyapunov*.

Xu y Li, presentan un algoritmo dinámico estocástico para fijar el precio óptimo para mejorar las ganancias desde la perspectiva del proveedor [21].

Por último en [22], Zaman y Grossu utilizan un algoritmo combinatorio basado en precios puntuales llamado *CA-Provision*. El algoritmo tiene tres fases: en la fase 1 se recolecta las ofertas de los usuarios, en la fase 2 se determina los usuarios ganadores y se asignan los recursos; y en la fase 3 se define el precio final que los usuarios ganadores deben pagar por las instancias solicitadas. La propuesta además asegura que un usuario no pagará por un precio menor al de un precio de reserva establecido por el proveedor.

IV. Consideraciones Varias

A. Métodos de estimación del tiempo de ejecución de una tarea en infraestructuras intermitentes

La estimación del tiempo de ejecución de cada tarea es un problema no trivial cuya resolución es necesaria para el problema de asignación de tareas en infraestructuras intermitentes.

En [13], Voorsluys et al. comparan los métodos indicados a continuación en relación al costo monetario y a la utilización de recursos.

- *Actual runtime*: El tiempo de la tarea se conoce a partir de un registro de tiempo de ejecución de las tareas.
- *Actual runtime with error*: Se basa en el *actual runtime* ligeramente modificado por un porcentaje aleatorio entre 0 y 10 %.
- *User supplied*: Asume que el tiempo de ejecución de la tarea es proveído por el cliente.
- *Fraction of User Supplied*: Se basa en que el método *User supplied* está sobre-estimado, por lo que se considera $\frac{1}{3}$ del tiempo original proveído por el usuario.
- *Recent Average*: Consiste en realizar un promedio del tiempo de ejecución de las dos últimas tareas completadas por el mismo usuario.

B. Estrategias de selección de oferta de instancias puntuales

Existen diferentes estrategias para la selección del valor de oferta para las instancias puntuales.

La utilización de una u otra estrategia podría significar mejores o peores resultados en la obtención de los recursos solicitados en un esquema de precios basado en subasta. Considerando esto, en el trabajo [12] se presenta un estudio y evaluación de las estrategias de ofertas indicadas a continuación:

- Ofertar el mínimo: Se oferta el valor mínimo observado en el histórico de precios más algún valor (de protección).
- Ofertar el promedio: Se oferta el precio promedio según el histórico.
- Ofertar el precio bajo demanda: Se oferta al precio de una instancia con esquema bajo demanda.
- Ofertar el precio más alto: Se oferta el precio más elevado observado en el histórico.
- Ofertar el precio actual: Se oferta el precio puntual actual más algún valor (de protección).

Voorsluys y Buyya proponen una técnica que tiene en cuenta un nivel de urgencia para cada tarea; éste se calcula en base a una estimación del tiempo de ejecución del trabajo y a la variación del precio puntual [12]. Asimismo, se comparan los métodos de estimación indicados y se observa el impacto que se tiene en el costo, violaciones del tiempo de finalización y la utilización del sistema.

C. Técnicas de tolerancia a fallos para infraestructuras intermitentes

Diferentes técnicas de tolerancia a fallo fueron estudiadas a fin de completar los trabajos respetando su tiempo de finalización y aumentando así la confiabilidad. A continuación se detallan las técnicas citadas en [12]:

- *Checkpointing*: consiste en guardar el estado de la ejecución de un trabajo para luego continuarlo a partir del último *checkpoint* antes del apagado de la instancia puntual. Se considera que el trabajo continuará una vez que la oferta original sea nuevamente mayor al precio puntual.
- *Task Migration*: Consiste en guardar constantemente todo el entorno de la ejecución de un trabajo en otra máquina virtual de manera a ponerlo en producción en una nueva instancia tan rápido como ocurra un fallo. Es muy parecido al esquema del *checkpointing* pero se diferencia en que se debe poner en producción tan pronto como haya aumentado el precio puntual.
- *Task duplication*: Se basa en la estimación del tiempo de ejecución de trabajos indicados en la Sección IV.A. Esta técnica crea una réplica de cada trabajo que se espera tenga una duración mayor a una hora.

D. Métricas de error para predicción de precios puntuales

Para la evaluación de las técnicas de predicción de precios puntuales, es necesaria la utilización de métricas apropiadas para el efecto.

El *Mean Absolute Percentage Error* (MAPE), utilizado en [19], es una métrica aceptada para la evaluación de técnicas de predicción [29]. El MAPE puede ser definido como:

$$MAPE_k = \frac{100}{N} \times \sum_{t=1}^{N} \left| \frac{F_{t+k} - A_{t+k}}{A_{t+k}} \right| \qquad (3)$$

dónde:

$MAPE_k$: Porcentaje de error absoluto en las predicciones del precio puntual para la instancia de tipo k;
F_{t+k}: Predicción del precio puntual;
A_{t+k}: Valor real del precio puntual;
N: Cantidad de puntos en el tiempo donde se realizó una predicción.

Alternativamente en el trabajo [16], se utiliza la función de error de una distribución Gaussiana, definida como:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (4)$$

V. Conclusiones y Trabajos Futuros

En la absoluta mayoría de los Centros de Datos, el bajo aprovechamiento de los recursos computacionales constituye un gran problema, ya que el promedio de utilización no supera el 10 %. Este desperdicio de capacidad implica además una gran ineficiencia energética. Amazon EC2 ha resuelto este problema comercializando su capacidad ociosa por medio de su oferta de instancias puntuales (*spot instances*).

Los clientes obtienen por medio del uso de instancias puntuales una gran flexibilidad en la ejecución de sus tareas. En cualquier momento los clientes pueden obtener los recursos computacionales que necesitan sin compromisos de pago a largo plazo. Al mismo tiempo, las instancias puntuales representan ahorros significativos para los clientes.

Sin embargo, las instancias puntuales presentan grandes desafíos tanto para proveedores como para clientes.

Para los proveedores, el precio fijado determina sus ingresos monetarios presentes y futuros. El precio puntual también determina que clientes serán atendidos y en que momento. Por lo tanto, los proveedores necesitan mecanismos que permitan determinar el precio puntual más adecuado.

Por otro lado, los clientes deben encontrar estrategias para determinar el monto de oferta más conveniente y el momento más apropiado para solicitar las instancias que necesitan. Los clientes también deben minimizar el impacto del apagado imprevisto de sus instancias debido a las fluctuaciones del precio puntual.

Uno de los requerimientos más importantes de un entorno de *Cloud Computing* es la provisión de una calidad de servicio confiable, la cual puede ser definida en términos de un acuerdo de nivel de servicio que describa características o métricas como el tiempo máximo de procesamiento de las tareas. De esta manera, los clientes tendrán la certeza de que sus requerimientos serán atendidos por los proveedores o intermediarios con los que esté trabajando.

Los intermediarios de servicios de *Cloud Computing* se encargan de las complejidades de uso de las instancias puntuales en nombre de sus clientes que así logran simplificar su operatoria cotidiana. De esta manera, toda la complejidad del sistema basado en instancias puntuales de precios variables se terceriza. Por otra parte, el bajo precio de los recursos puntuales puede representar para el intermediario una oportunidad de obtener beneficios económicos adicionales.

En el presente trabajo se han estudiado y seleccionado los trabajos más representativos del estado del arte. A partir de este estudio, se propone por primera vez una taxonomía de la provisión de recursos en un esquema basado en subasta, con los siguientes criterios de clasificación: (1) la perspectiva del proveedor o del usuario, (2) el problema resuelto, (3) el enfoque de optimización, (4) las funciones objetivo y (5) las técnicas de solución.

Los trabajos seleccionados han sido clasificados de acuerdo a la taxonomía propuesta. En base a esto se han observado con claridad las áreas más estudiadas y se han identificado las oportunidades de investigación existentes en esta interesante área. En este sentido, en primer lugar, se podría trabajar en mejorar la formulación de algunos de los objetivos, volviéndolos más realistas. Un ejemplo claro lo constituye el objetivo de maximización de la confiabilidad, el cual podría atender varios niveles de SLA (e.g. críticos, opcionales, entre otros).

Otro aspecto que puede ser profundizado es el enfoque utilizado para la solución de los problemas. En la Tabla I se puede observar que en la mayoría de los casos solo se proponen enfoques mono-objetivo. Solamente para el problema de asignación de tareas en infraestructuras intermitentes se ha encontrado una formulación que considera la minimización del tiempo total de proceso y del costo como un problema multi-objetivo. De esta manera, queda trabajo por realizar en formular los problemas considerando simultáneamente varios de los objetivos identificados. La aplicación de un enfoque multi-objetivo se podría combinar con algoritmos evolutivos tales como: ACO, PSO, SA y HS por citar algunos.

Otro potencial tema de investigación identificado es resolver el problema de selección de valor de oferta para instancias puntuales utilizando algoritmos evolutivos para la predicción del precio puntual, como por ejemplo la programación genética lineal [30]. En este contexto, se propone como trabajo futuro la evaluación de otras métricas de error para predicción de precios puntuales que puedan representar de forma adecuada la utilidad de los valores predecidos considerando que una predicción menor del precio puntual, por más mínimo que sea, implicaría la no provisión de los recursos solicitados.

En los trabajos estudiados también se nota que técnicas de solución como las meta-heurísticas bio-inspiradas no han sido consideradas. Varios trabajos futuros podrían estar basados en la utilización de dichas técnicas para resolver el problema de la asignación de tareas en instancias puntuales, comparando su efectividad y escalabilidad con respecto a las técnicas ya estudiadas en el estado del arte.

Además, se destaca como trabajo futuro el desarrollo de servicios que ayuden a los usuarios en la adopción de *Cloud Computing* en los actuales mercados. Una estrategia válida para esto sería un servicio como Cloudorado [31], pero que a diferencia de este, se enfoque en servicios de estrategias de selección del valor de oferta para instancias puntuales de manera a facilitar la contratación de recursos en esquemas basados en subasta, aprovechando de esta manera el bajo costo de los mismos y mitigar los riesgos existentes con relación a la confiabilidad de estos recursos.

Finalmente, se propone como trabajo futuro el estudio de posibles modelos de intermediarios basados en esquemas híbridos de tipos de instancias como los estudiados en [32] (para instancias reservadas y bajo demanda) y en [14] (para instancias puntuales y bajo demanda). En este caso, se podría contar con un modelo de intermediario que considere según sea necesario la contratación de instancias reservadas o bajo demanda de manera a aumentar la confiabilidad de las instancias puntuales, considerando las ventajas existentes entre los diferentes tipos de instancia que ofrece Amazon EC2 para cada contexto de utilización [4].

REFERENCIAS BIBLIOGRÁFICAS

[1] R. Buyya, C. S. Yeo, y S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities" en *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, páginas. 5–13, Ieee, 2008.

[2] M. Hogan, F. Liu, A. Sokol, y J. Tong, "Nist cloud computing standards roadmap" *NIST Special Publication*, vol. 35, 2011.

[3] L. Leong, D. Toombs, B. Gill, G. Petri, y T. Haynes, "Magic quadrant for cloud infrastructure as a service" *Gartner, http://www.gartner.com/technology/reprints.do*, 2014.

[4] A. W. Services, "Instancias puntuales de Amazon EC2" *http://aws.amazon.com/ec2/spot-instances/*, Mar 2015.

[5] E. Star, "Consolidation of lightly utilized servers" *http://www.energystar.gov/*, Apr 2015.

[6] Y. Song, M. Zafer, y K.-W. Lee, "Optimal bidding in spot instance market" en *INFOCOM, 2012 Proceedings IEEE*, páginas. 190–198, Mar 2012.

[7] S. Yi, A. Andrzejak, y D. Kondo, "Monetary cost-aware checkpointing and migration on amazon cloud spot instances" *Services Computing, IEEE Transactions on*, vol. 5, no. 4, páginas. 512–524, 2012.

[8] I. Jangjaimon y N.-F. Tzeng, "Effective cost reduction for elastic clouds under spot instance pricing through adaptive checkpointing" *IEEE Transactions on Computers*, vol. 99, no. PrePrints, página. 1, 2013.

[9] S. Yi, D. Kondo, y A. Andrzejak, "Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud" en *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, páginas. 236–243, IEEE, 2010.

[10] A. W. Services, "New EC2 spot instance termination notices" *https://aws.amazon.com/blogs/aws/new-ec2-spot-instance-termination-notices/*, Mar 2015.

[11] S. V. Sevastianov y G. J. Woeginger, "Makespan minimization in open shops: A polynomial time approximation scheme" *Mathematical Programming*, vol. 82, no. 1-2, páginas. 191–198, 1998.

[12] W. Voorsluys y R. Buyya, "Reliable provisioning of spot instances for compute-intensive applications" en *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, páginas. 542–549, IEEE, 2012.

[13] W. Voorsluys, S. K. Garg, y R. Buyya, "Provisioning spot market cloud resources to create cost-effective virtual clusters" en *Algorithms and Architectures for Parallel Processing*, páginas. 395–408, Springer, 2011.

[14] A. Vintila, A.-M. Oprescu, y T. Kielmann, "Fast (re-)configuration of mixed on-demand and spot instance pools for high-throughput computing" en *Proceedings of the First ACM Workshop on Optimization Techniques for Resources Management in Clouds*, ORMaCloud '13, (New York, NY, USA), páginas. 25–32, ACM, 2013.

[15] S. Tang, J. Yuan, y X.-Y. Li, "Towards optimal bidding strategy for amazon ec2 cloud spot instance" en *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, páginas. 91–98, Jun 2012.

[16] B. Javadi, R. Thulasiramy, y R. Buyya, "Statistical modeling of spot instance prices in public cloud environments" en *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, páginas. 219–228, Dec 2011.

[17] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, y D. Tsafrir, "Deconstructing amazon ec2 spot instance pricing" *ACM Trans. Econ. Comput.*, vol. 1, páginas. 16:1–16:20, Sept. 2013.

[18] B. Javadi, R. K. Thulasiram, y R. Buyya, "Characterizing spot price dynamics in public cloud environments" *Future Generation Computer Systems*, vol. 29, no. 4, páginas. 988–999, 2013.

[19] V. K. Singh y K. Dutta, "Dynamic price prediction for amazon spot instances" en *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, páginas. 1513–1520, Jan 2015.

[20] P. Wang, Y. Qi, D. Hui, L. Rao, y X. Liu, "Present or future: Optimal pricing for spot instances" en *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, páginas. 410–419, IEEE, 2013.

[21] H. Xu y B. Li, "Dynamic cloud pricing for revenue maximization" *Cloud Computing, IEEE Transactions on*, vol. 1, páginas. 158–171, Jul 2013.

[22] S. Zaman y D. Grosu, "A combinatorial auction-based mechanism for dynamic vm provisioning and allocation in clouds" *Cloud Computing, IEEE Transactions on*, vol. 1, no. 2, páginas. 129–141, 2013.

[23] C. C. Coello, G. B. Lamont, y D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer Science & Business Media, 2007.

[24] M. Mazzucco y M. Dumas, "Achieving performance and availability guarantees with spot instances" en *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*, páginas. 296–303, IEEE, 2011.

[25] B. Barán y M. Schaerer, "A multiobjective ant colony system for vehicle routing problem with time windows." en *Applied Informatics*, páginas. 97–102, 2003.

[26] J. Lima y B. Barán, "Optimización de enjambre de partículas aplicada al problema del cajero viajante bi-objetivo" *Revista Iberoamericana de Inteligencia Artificial, Asociación Española para la Inteligencia Artificial Valencia España*, páginas. 67–76, 2006.

[27] H. Meyer y B. Barán, "Una nueva propuesta de templado simulado multi-objetivo" en *Computing Conference (CLEI), 2009 XXXV Latin American*, páginas. 1–8, 2009.

[28] J. Ricart, G. Hüttemann, J. Lima, y B. Barán, "Multiobjective harmony search algorithm proposals" *Electronic Notes in Theoretical Computer Science*, vol. 281, páginas. 51–67, 2011.

[29] J. Armstrong y F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons" *International Journal of Forecasting*, vol. 8, no. 1, páginas. 69 – 80, 1992.

[30] R. Sanchez, J. Martinez, y B. Barán, "Macro-economic time-series forecasting using linear genetic programming" en *11th Joint International Conference on Information Sciences*, Atlantis Press, 2008.

[31] Cloudorado, "Cloudorado: Cloud computing price comparison engine" *https://www.cloudorado.com/*, May 2015.

[32] S. Nesmachnow, S. Iturriaga, y B. Dorronsoro, "Efficient heuristics for profit optimization of virtual cloud brokers" *Computational Intelligence Magazine, IEEE*, vol. 10, no. 1, páginas. 33–43, 2015.