VOL. 5 Nº1 - 2023

IBARRA - ECUADOR

Facultad de Ingeniería en **Ciencias Aplicadas**

Innovation & Development in Engineering and Applied Science

IDEAS

Invocación de métodos remotos (RMI): una revisión de la literatura.

http://doi.org/110.53358/ideas.v5i1.869

Margoth Guaraca[1], José Jácome[2], Alexander Pinchao[2], Jonathan Silva[2], Fernando Imbaquingo[2]

[1] Universidad do Minho, Braga, Portugal

[2] Universidad Técnica del Norte, Ibarra, Ecuador

[1]meguaraca@espe.edu.ec, [2]{jgjacome, appinchaop, jxsilvam, efimbaquingoe}@utn.edu.ec

Fecha de envío, marzo 24/2023 - Fecha de aceptación, mayo 19/2023 - Fecha de publicación, Julio 10/2023

Resumen. La revisión de la literatura sigue siendo importante para comprender el acontecimiento de diferentes estudios sobre un tema específico y de interés para la sociedad, es por ello que, se ha visto la necesidad de hacer una revisión literaria referente al mecanismo de invocación de métodos remotos. Remote Method Invocation (RMI) como tal, es una tecnología utilizada para la comunicación entre objetos, un tema que en particular ha tenido mucha relevancia en computación distribuida y procesamiento en paralelo, lo que ha permitido el desarrollo de múltiples aplicaciones empresariales al igual que juegos. Por ello, teniendo como guía una metodología para la revisión bibliográfica y la gestión de información sobre temas científicos que se basa en centralizar un problema, buscar y organizar información, se ha iniciado por seleccionar un determinado número de investigaciones para proceder con la revisión de literatura, de tal forma que, se pueda extraer conceptos y aportes más importantes de cada una de las investigaciones. Se realizó una lista de temas que engloban RMI como base de estudio y se buscó lo más relevante para realizar en principio, una introducción referente a computación distribuida, RMI y su forma de implementación, además, a través de tablas, se especificó las definiciones de algunos autores, así como las ventajas y desventajas que presenta esta tecnología en cuanto a su implementación y uso en el área de Ciencias de la Computación.

Palabras Clave: Java, Distributed systems, Remote Method Invocation, object, parallel processing.

Abstract. The literature review holds significant importance in comprehending the existing studies on a specific topic, which are of great interest to society. Therefore, it is imperative to conduct a literature review concerning the mechanism of remote method invocation (RMI). RMI is a technology utilized for object communication, particularly relevant in the fields of distributed computing and parallel processing. Its applications have played a crucial role in the development of numerous business applications and games. To conduct this literature review effectively, it has been adopted a methodology for information management on scientific topics, which involves centralizing the problem, conducting a thorough search, and organizing the information. Initially, it was selected a set of investigations as the basis for the literature review, aiming to extract the most important concepts and contributions from each study. It has been compiled a list of topics encompassing RMI as the foundation for this study and focused on the most relevant aspects. As a starting point, it was provided an introduction to distributed computing, RMI, and its implementation methodology. Furthermore, were employed tables to present definitions from various authors and outlined the advantages and disadvantages of this technology within the realm of Computer Science.

Keywords: Java, Distributed systems, Remote Method Invocation, object, parallel processing.

Autor de correspondencia:

Margoth Guaraca, meguaraca@espe.edu.ec

Introducción

Las aplicaciones que se van desarrollando cada vez cumplen con más funcionalidades, por lo cual requieren un mayor procesamiento, en especial, las que utilizan inteligencia artificial y aprendizaje automático. Es debido a ello que, se hace uso de la computación distribuida para un mejor rendimiento de las aplicaciones [1].

RMI es una técnica de computación distribuida que divide una tarea en múltiples subprocesos para ejecutarlos simultáneamente en distintas máquinas [2], esto gracias a su enfoque basado en la distribución de objetos y la transportación de los mismos a través de la red, permitiendo que se pueda invocar métodos de los objetos almacenados en otra máquina, sin la necesidad de estar en el mismo lugar [3], también se hace uso de un marco de capa de red, el cual contribuye a un mejor procesamiento en paralelo [4].

RMI es un conocido trabajo de distribución de middleware de procesamiento paralelo [5], utiliza una de las formas más rápidas de conexión entre el cliente y el servidor, y asegura una conexión confiable, puesto que, una solicitud puede pasar a través del firewall [6], realiza serialización de objetos y aplica polimorfismo orientado a objetos [7]. RMI utiliza la capa de transporte, de referencia remota y la capa de aplicaciones, con el fin de tener la comunicación entre los sistemas maestro y esclavo [20].

RMI estándar utiliza una implementación de objeto remoto exportada [8], esto significa que, un objeto de un sistema puede llamar a un método en un objeto en otro lugar de una red, permitiendo así envío de parámetros y la obtención de resultados automáticamente [9], mediante su API se realiza la invocación de un método de un objeto remoto por un objeto residente local [10].

Para aplicaciones que implementa RMI, a menudo cuentan con dos programas independientes como mínimo, que en este caso son el cliente y el servidor [11], los cuales permiten cumplir con el principio de RMI que es el llamado o la invocación de métodos remotos y el envío de parámetros [12]. Hay que tomar en cuenta que en una implementación eficiente de RMI, su desarrollo hace posible descargar clases remotas en una aplicación en ejecución, lo que causa una sobrecarga de tiempo de ejecución importante [13], sin embargo, RMI proporciona compatibilidad para referencias a objetos punto a punto mediante TCP [14].

RMI se presenta como un servicio que proporciona una base sólida y segura para la implementación en varias aplicaciones y escenarios dentro de una red, incluido aplicaciones que requieran una gran cantidad de datos de entrada [15]. Se puede encontrar aplicaciones como ActiveRMI que implementa el servicio de RMI [16], los servidores de juegos están diseñados para ser reutilizables en cualquier otro proyecto que requiera un sistema informático distribuido, debido a la implementación de esta tecnología, que permite a los desarrolladores concentrarse en el diseño del modelo de objetos y la lógica de la aplicación, en lugar de implementar la conexión de socket entre el servidor y el cliente [17].

Una de las principales razones porque se han implementado en muchas ocasiones RMI, se debe a que es una plataforma que ya lleva sus años y es madura en cuanto a la computación distribuida se refiere, además de tener suficiente documentación, lo que les brinda a los desarrolladores una confianza para su uso [18].

También se ha demostrado que el sistema de clúster basado en RMI es mucho más eficiente que la programación de Socket, esto en base a resultados experimentales que se han obtenido de algunos estudios realizados, en donde se ha evidenciado que, al realizar un procesamiento con RMI, requiere menos tiempo de ejecución que la programación basada en Socket, además, el uso directo de la programación de Socket no es muy capaz de enviar grandes cantidades de datos en una sola instancia a todo el clúster [19].

Se puede aplicar RMI para la comunicación entre diferentes sistemas en un clúster, en donde se puede implementar el protocolo de método remoto, que funciona a través del protocolo de Transmission Control Protocol/Internet Protocol (TCP / IP). Los sistemas distribuidos basados en RMI pueden funcionar directamente sin almacenar datos en los archivos a medida que suministra el servicio de conexión. RMI proporciona el establecimiento de conexión confiable entre el cliente y el servidor [20]. Además, se puede utilizar para construir un servicio de almacenamiento en caché parcial, puesto que los objetos se pueden exportar y recuperar a través de la red. Luego, se pueden invocar llamadas a métodos en el servidor de objetos remotos, lo que resulta en una ejecución remota [22].

RMI, si bien es un mecanismo de arquitectura distribuida y ofrecido por Java, esto se puede aplicar también con el uso de máquinas virtuales, en la que los métodos de objetos remotos pueden invocarse desde otras máquinas virtuales Java, posiblemente ubicadas en diferentes hosts. [23]. La tecnología Java RMI permite llamar a un método Java en una máquina virtual remota y, es fácilmente aplicable a una red informática heterogénea. Uno de los requisitos cruciales es, poder usarlo para varios tipos de computadoras con varios sistemas operativos instalados. Esto es relativamente fácil con RMI, puesto que está incluida en Java, que es un lenguaje de programación independiente de la plataforma [24]. Los estudios realizados sugieren usar RMI que generen y operen con objetos Java en un sistema Remoto [23].

El aplicar los conceptos de RMI en plataformas móviles puede traer varios beneficios que son la mejor transparencia de recursos del dispositivo cuando se pueden asignar múltiples recursos para una llamada de función. Sin embargo, la tecnología RMI original no admite ningún enrutamiento que no sea el que admiten las capas de red subyacentes; si no hay redes entre dos dispositivos (cliente y servidor), un objeto en el primer dispositivo no puede invocar un método para un objeto en el segundo dispositivo. Además, RMI es una tecnología obsoleta que depende en gran medida del paquete javax de la biblioteca Java SDK; por lo tanto, no es compatible con plataformas móviles como Android [25]. Es por ello que, se dispone de Android RMI que proporciona una interfaz de programación de aplicaciones (API) simple a nivel de usuario y sus abstracciones son casi idénticas a las utilizadas para invocar un servicio dentro del mismo dispositivo. Esto evita que los usuarios modifiquen los códigos fuente de las aplicaciones existentes cuando se ejecutan en múltiples dispositivos Android, y también son portátiles. A diferencia de los enfoques anteriores en los que solo se pueden invocar servicios de aplicaciones, Android RMI permite a los usuarios invocar servicios del sistema y servicios de aplicaciones entre dispositivos mediante el uso de un formato de paquete remoto, que es un mensaje extendido desde el formato de datos Parcelable (una interface para clases, cuyas instancias se pueden escribir y restaurar desde una clase tipo Parcel). Android RMI está diseñado en base a tres objetivos de diseño importantes: transparencia, portabilidad y rendimiento [26].

Materiales y Métodos

La revisión literaria empieza por definir las bases de datos científicas y bibliográficas en las que se hará la búsqueda del tema que se pretende investigar, en el caso planteado "RMI (Método de Invocación Remoto)", para ello se consideró las siguientes bases de datos bibliográficas: Scopus, IEEE Explore, Google Scholar debido a que se enfocan en la investigación científica y pueden ser de utilidad para la investigación.

Una vez identificadas las bases de datos bibliográficas se procede a hacer la búsqueda con las siguientes cadenas de texto: "RMI", "Remote Method Invocation".

Empezando con IEEE Explore, de donde se obtuvo un resultado preliminar de 417 documentos, entre conferencias, revistas científicas y artículos, en esta parte se realizó una lectura del resumen de los artículos, priorizando los publicados recientemente, con el objetivo de encontrar los documentos más relevantes que aporten información a la investigación en cuestión, obteniendo finalmente 30 que se consideraron los más importantes.

Asimismo, se procedió a hacer la búsqueda en Scopus con un resultando inicial de 3423 documentos, se aplicó un filtro, especificando las áreas temáticas como ciencias de la computación, ingeniería y computación distribuida, además de limitar el tipo de documento como artículo, se logró reducir la cantidad de documentos a 123 y finalmente se determinó que 24 eran los más relevantes.

Por último, en la base de datos de Google Scholar, que en una búsqueda inicial con las palabras claves se obtuvo 25000 resultados, mediante un filtro se determinó una cantidad de 80 documentos que contenían el tema de interés y finalmente se seleccionó 21 considerados como los más importantes.

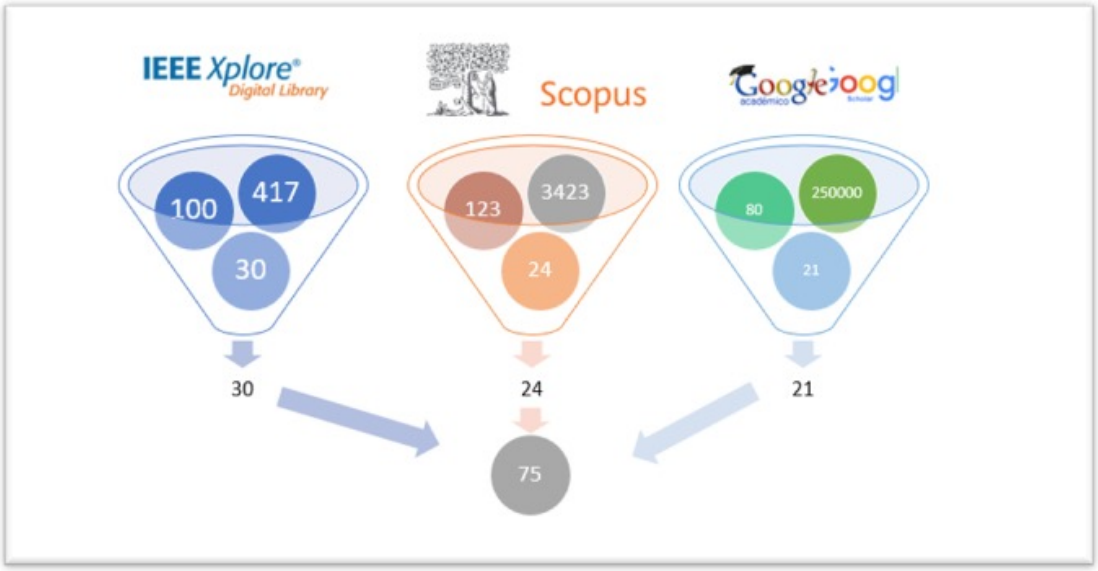

Gráfico 1 Filtro de documentos. Elaboración propia

Luego de la búsqueda de los documentos se realizó la lectura comprensiva de los mismos, para determinar las definiciones de los distintos autores y sintetizar en una tabla, además de las ventajas y desventajas que presenta RMI como tal.

Resultados y Discusión

De la revisión de literatura realizada se obtuvieron algunas definiciones de RMI, las mismas que son conceptualizadas por los autores con sus propias palabras, y que tenían concordancia en su significado. Estas definiciones se detallan a continuación en una tabla con su autor correspondiente.

Tabla 1 – Definición RMI según autores

Autor	RMI
Fujie Gao	Sistema complejo, que incluye fenómenos como la interacción de la interfaz de choque, la inestabilidad hidrodinámica, el flujo multifásico y la mezcla turbulenta [27].
Ahmet Sayar	Tipo de llamada a procedimiento remoto que es independiente de la red, ligera y totalmente portátil [28].
Daniel Tejera	Se basa en un hilo que llama a un método en una computadora remota, en donde un servidor obtiene esta invocación, ejecuta el método apropiado y devuelve los resultados del método y el cliente espera hasta que se reciba esta información [29].
Sachin Bagga	Es una forma factible para llamar a un objeto que reside en alguna máquina remota [2].
Minh Le	Es una biblioteca y un marco de nivel relativamente bajo que se basa en llamadas a métodos [30].
Sachin Bagga	Método que permite la comunicación entre los sistemas maestro y esclavo, y proporciona un servicio de establecimiento de conexión confiable que incluso puede pasar a través del firewall [20].
Jan Vimmr	Tecnología que permite llamar a un método en una máquina virtual remota y, es fácilmente aplicable a una red informática heterogénea [24].
[25]	Es una arquitectura distribuida en la que los métodos de objetos remotos pueden invocarse desde otras máquinas posiblemente ubicadas en diferentes hosts [25].
Minh Ahuja	Es una tecnología para distribuir objetos a través de la red, además, permite invocar llamadas de método en objetos que residen e n otra máquina sin tener que mover esos objetos a la máquina que hace la llamada [3].
D.-A. German	Significa que un objeto de un sistema puede llamar a un método en un objeto en otro lugar de una red. Enviará parámetros y obtendrá un resultado de vuelta automáticamente [9].
M. Govindaraju	Es una API para la invocación de métodos remotos: la invocación de un método en un objeto remoto por un objeto residente local [10].
K. Bergner	Es una arquitectura que permite la comunicación entre objetos en procesos diferentes y espacios de direcciones, posiblemente en diferentes anfitriones [31].
Diya Sharon	Es una forma ideal y poderosa de manipular objetos presentes en otro sistema con mucha facilidad [32].

Fuente: Elaboración Propia

Con las distintas definiciones encontradas se determinó una definición general de RMI que se presenta a continuación:

Se define a RMI como una tecnología muy ligera utilizada en computación distribuida, que posibilita la comunicación entre objetos, permitiendo la realización de llamadas a métodos remotos, independientemente de la red, esto mediante un cliente que realiza la llamada y un servidor que devuelve una respuesta automáticamente.

Además, de las definiciones mencionadas anteriormente, se logra resaltar algunas ventajas y desventajas, evidenciadas en los documentos referentes a RMI. Estas se presentan en la siguiente tabla:

Tabla 2 – Ventajas y desventajas de RMI

Ventajas	Desventajas
Puede ejecutarse sobre cualquier máquina virtual de Java [24].	Requiere que los desarrolladores lo configuren y mantengan manualmente [30].
Puede pasar a los firewalls [6].	Un servidor que acepta ciegamente cualquier solicitud de RMI se abre a ataques de sobrecarga de memoria y / o computacionales [15].
Tiene su propio protocolo de trabajo [21].	La RMI nativa provoca una reducción en la velocidad de rendimiento [32].
Tiene una disposición que puede restablecer la conexión. [21].	Su política de diseño y especificaciones para un entorno multi cliente que controla dispositivos físicos de un sistema de control / monitoreo del acelerador [33].
RMI es mucho más eficiente que la programación de Socket ya que requiere menos tiempo de ejecución [19].	Tiempo de bloqueo inaceptable debido al sincronismo de RMI [33].
Es una plataforma que ya lleva sus años y es madura en cuanto a la computación distribuida se refiere [18].	Tiempo de bloqueo inaceptable para un cliente / servidor cuando se produce una falla de red [33].

Fuente: Elaboración Propia

Aunque esta tecnología brinda algunas ventajas, en la actualidad ya no es de uso común, debido a la aparición de nuevos métodos de comunicación como por ejemplo las API REST, por lo cual está quedando obsoleta. Pero es un buen referente para estudiar y analizar la computación distribuida desde su inicio.

Conclusiones

La revisión literaria es un aporte muy importante para futuras investigaciones porque da una reseña del estado actual de un tema específico y agiliza procesos de investigaciones posteriores, con mayor énfasis en temas de tecnología como RMI, puesto que, son áreas que están en constante evolución con mejoras continuas.

RMI fue uno de los pilares fundamentales de la computación distribuida para pasar a lo que hoy existe, como son los servicios en la nube, muy eficientes y de gran utilidad para las aplicaciones empresariales, que siguen siendo algo que se percibe a diario y se habla mucho en las empresas como una solución tecnológica al consumo de servicios en la nube.

De igual manera, en la actualidad existen paquetes y diferentes abstracciones de software como Android RMI para el desarrollo de aplicaciones móviles, que permiten invocar métodos remotos entre varios dispositivos, lo que reduce el tiempo de invocación y tamaño en bytes que se traduce en una mejora en el rendimiento.

Referencias

1. Cooke, Ryan; Fahmy, Suhaib; (2020). "A Model for Distributed In-Network and Near-Edge Computing With Heterogeneous Hardware". Future Generation Computer Systems. Recuperado de: https://www.sciencedirect.com/science/article/pii/S0167739X19312130?via%3Dihub

2. Bagga, Sachin; Girdhar, Akshay; Chandra, Munesh; (2017). "SPMD based time sharing intelligent approach for image denoising". Journal of Intelligent and Fuzzy Systems. Recuperado de: https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs169292

3. S.P. Ahuja; R. Quintao; (2002). " Performance evaluation of Java RMI: a distributed object architecture for Internet based applications". IEEE. San Francisco. Recuperado de: https://ieeexplore.ieee.org/document/876585

4. Musab, Ali; Shehab, Hamood; Najm, Ihab; Al-yousif, Shahad; Tahir, Nooritawati; (2019). "A Comparative Study in Remote Techniques and Event-Based Invocations". IEEE 10th Control and System Graduate Research Colloquium. Recuperado de: https://ieeexplore.ieee.org/document/8837063

5. M.A. de Miguel; (2002). " Solutions to make Java-RMI time predictable". IEEE. Magdeburgo. Recuperado de: https://ieeexplore.ieee.org/document/922862

6. Harmanpreet Kaur ; Sachin Bagga ; Ankit Arora (2015). " RMI approach to cluster based Winograd's variant of Strassen's method". IEEE. Amritsar. Recuperado de: https://ieeexplore.ieee.org/document/7375307

7. Yu Weihong (2009). " Communication Mechanism of Search and Rescue at Sea Intelligent Decision Support System Based on RMI". IEEE. Phuket. Recuperado de: https://ieeexplore.ieee.org/document/4737118

8. Rudiger Kapitza ; Jorg Domaschka ; Franz J. Hauck ; Hans P. Reiser ; Holger Schmidt (2006). "FORMI: integración de objetos fragmentados adaptativos en Java RMI". IEEE. Recuperado de: https://ieeexplore.ieee.org/document/4012576

9. D.-A. German; (2005). " RMI: observing the distributed pattern". IEEE. USA. Recuperado de: https://ieeexplore.ieee.org/document/1408619

10. M. Govindaraju ; A. Slominski ; V. Choppella ; R. Bramley ; D. Gannon; (2006). " Requirements for and Evaluation of RMI Protocols for Scientific Computing". IEEE. Dallas. Recuperado de: https://ieeexplore.ieee.org/document/1592774

11. Arabi E. Keshk (2006). "Implementation of Distributed Application using RMI Java threads ". IEEE. Guiza. Recuperado de: https://ieeexplore.ieee.org/document/4458214

12. D.-A. German; (2004). " The net worth of an object-oriented pattern: practical implications of Java RMI". IEEE. Newport Beach. Recuperado de: https://ieeexplore.ieee.org/document/1316118

13. Sudarsan, V; Sugumar, R; (2018). "Building a distributed K-Means model forWeka using remote method invocation (RMI) feature of Java". Wiley. Recuperado de: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5313

14. M. Sharp; A. Rountev (2006). " Static Analysis of Object References in RMI-Based Java Software". IEEE. Moscu. Recuperado de: https://ieeexplore.ieee.org/document/1707666

15. Król, Michał; Habak, Karim; Oran, David; Kutscher, Dirk; Psaras, Ioannis; (2018). "RICE: remote method invocation in ICN". ICN 2018 - Proceedings of the 5th ACM Conference on Information-Centric Networking. Recuperado de: https://dl.acm.org/doi/10.1145/3267955.3267956

16. Meng-Chun Wueng ; Fu-Fang Yang ; Cheng-Zen Yang; (2005). " A novel Java RMI middleware design for active networks". IEEE. Thailand. Recuperado de: https://ieeexplore.ieee.org/document/1414708

17. Artur Mlodzinski ; Jaroslaw Wozniak ; Wojciech Zabierowski ; Andrzej Napieralski; (2007). " Portal de juegos en línea como un ejemplo del uso de la tecnología J2EE y RMI". IEEE. Lviv-Polyana. Recuperado de: https://ieeexplore.ieee.org/document/4297629

18. Abhaya Induruwa ; Mona Christian; (2008). " Building Sensor Networks with Distributed Intelligence Using Java RMI". IEEE. Cap. Esterel. Recuperado de: https://ieeexplore.ieee.org/document/4622670

19. Bagga, Sachin; Girdhar, Akshay; Trivedi, Munesh; Bao, Yinan; Du, Jingwen; (2018). "RMI Approach to Cluster Based Image Decomposition for Filtering Techniques". Advances in Intelligent Systems and Computing. Recuperado de: https://link.springer.com/chapter/10.1007%2F978-981-10-3773-3_38

20. Bagga, Sachin; Girdhar, Akshay; Yan, Rajun; Lin, Zihan; (2016). "Virtualization Approach to Cluster Based Winograd's Variant of Strassen's Method using RMI". Second International Conference on Computational Intelligence & Communication Technology. Recuperado de: https://ieeexplore.ieee.org/document/7546579.

21. Bagga, Sachin; Girdhar, Akshay; Chandra, Munesh; Yang Trivedi; (2016). "RMI Approach to Cluster Based Cache Oblivious Peano Curves". Proceedings - 2016 2nd International Conference on Computational Intelligence and Communication Technology. Recuperado de: https://ieeexplore.ieee.org/document/7546580

22. Gascon-Samson, Julien; Coppinger, Michael; Jin, Fan; Kienzle, Jörg; Kemme, Bettina; (2017). "CacheDOCS: A Dynamic Key-Value Object Caching Service". Proceedings - IEEE 37th International Conference on Distributed Computing Systems Workshops, ICDCSW 2017. Recuperado de: https://ieeexplore.ieee.org/document/7979851

23. Boon-Hee Kim ; Young-Chan Kim (2002). " LORB: Infrastructure Design of RMI System for Lightweight Object Request Broker". IEEE. Corea del Sur. Recuperado de: https://ieeexplore.ieee.org/document/1032563

24. Vimmr, Jan; Bublík, Ond rej; Pecka, Aleš; (2017). " A parallel implementation of an implicit discontinuous Galerkin finite element scheme for fluid flow problems". Advances in Engineering Software. Recuperado de: https://www.sciencedirect.com/science/article/abs/pii/S0965997816306858?via%3Dihub

25. Le, Minh; Clyde, Stephen; Kwon, Young-Woo; (2019). "Enabling multi-hop remote method invocation in device-to-device networks". Human-centric Computing and Information Sciences. Recuperado de: https://hcis-journal.springeropen.com/articles/10.1186/s13673-019-0182-9

26. Kang, HeeEun; Jeong, Kihyun; Lee, Kwonyong; Park, Sungyong; Kim, Youngjae; (2016). "Android RMI: a user-level remote method invocation mechanism between Android devices". Journal of Supercomputing. New York. Recuperado de: https://link.springer.com/article/10.1007%2Fs11227-015-1471-3

27. Gao, Fujie; Zhang, Yousheng; He, Zhiwei; Tian, Baolin; (2016). "Formula for growth rate of mixing width applied to Richtmyer-Meshkov instability". Physics of Fluids. Recuperado de: https://aip.scitation.org/doi/10.1063/1.4966226

28. Sayar, Ahmet. (2015). "Ant-based interactive workflow management: a case study on RMI". International Journal of Communication Systems. Recuperado de: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.2766

29. Tejera, Daniel; Tolosa, Ruth; Miguel, A; Alonso, Alejandro; (2010). "Two Alternative RMI Models for Real-Time Distributed Applications". ISORC 2010 - 2010 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing. Madrid. Recuperado de: https://ieeexplore.ieee.org/document/1420996

30. Le, Minh; Clyde, Stephen; (2018). "INGRIM: A Middleware to Enable Remote Method Invocation Routing in Multiple Group Device-to-Device Networks". Proceedings - IEEE 2018 International Congress on Cybermatics. Recuperado de: https://ieeexplore.ieee.org/document/8726609

31. K. Bergner ; A. Rausch ; M. Sihling; (2002). " Casting an abstract design into the framework of Java RMI". IEEE. Dunedin. Recuperado de: https://ieeexplore.ieee.org/document/707661

32. Sharon, Diya; Niranjani, T; (2015). "Effective usage of customizable extensions on JAVA for optimized high performance computation". Proceedings - NCET NRES EM 2014: 2nd IEEE National Conference on Emerging Trends in New and Renewable Energy Sources and Energy Management. Recuperado de: https://ieeexplore.ieee.org/document/7088744

33. Noriichi, Kanaya; Mori, Shintaro; Shikanai, Akihiko; (2016). "Asynchronous distributed object model using Java for the control system of a synchrotron radiation source". IEEE Transactions on Nuclear Science. Recuperado de: https://ieeexplore.ieee.org/document/7579226