Integración de la IA Generativa en Sistemas Distribuidos

David Bacas Posadas[1], Julian Carrión Tovar[1], Minerva Cebrián Marín[1], Miguel Ángel Luque Gómez[1], and Pablo Hernández Ibáñez[1]

University of Granada, Granada, Spain

Abstract. Este trabajo analiza la integración de la Inteligencia Artificial Generativa (GenAI) en arquitecturas de sistemas distribuidos, fundamentándose en los principios de diseño de Coulouris et al.. Debido al tamaño masivo de los modelos de lenguaje (LLMs), la ejecución en dispositivos individuales resulta inviable, exigiendo soluciones distribuidas para superar el "muro de la memoria". Se exploran ventajas críticas como la escalabilidad horizontal extrema y la eficiencia mediante algoritmos de gestión de memoria como PagedAttention. Asimismo, se examinan inconvenientes técnicos, incluyendo la latencia de red y el impacto de los nodos lentos o stragglers que pueden ralentizar el entrenamiento en un 42.5%. El estudio detalla aplicaciones industriales reales a través de plataformas como T-Systems y orquestadores como Kubernetes y Ray. Finalmente, se proponen cuestiones técnicas para el seminario académico del 26 de marzo, abordando la tolerancia a fallos y la soberanía del dato en entornos federados. Este marco proporciona una visión técnica y ética alineada con las directrices proporcionadas para la asignatura de Diseño de Sistemas Distribuidos de la Universidad de Granada.

Keywords: GenAI · Sistemas Distribuidos · vLLM · Chatbots · Escalabilidad.

1 Introducción y motivación del trabajo

La integración de la IA Generativa en sistemas distribuidos es un pilar de la ingeniería informática moderna. Debido a que los modelos actuales alcanzan trillones de parámetros, su computación en dispositivos únicos es técnica y económicamente inviable, exigiendo arquitecturas distribuidas para su despliegue.

1.1 Motivación técnica

El "muro de la memoria" es el principal motor de este estudio. Ante el desfase entre la capacidad de cómputo de las GPUs y el ancho de banda de su memoria, la inferencia de LLMs se ha vuelto una tarea *memory-bound*. Esto obliga a fragmentar los modelos mediante paralelismo de tensores y tuberías para operar en tiempo real con latencias aceptables.

1.2 Motivación académica

Académicamente, este trabajo responde a la necesidad de gestionar arquitecturas complejas bajo estándares éticos. El auge de la *Edge AI* y la demanda de soberanía digital impulsan la investigación hacia soluciones colaborativas que garanticen respuestas instantáneas y privacidad, evitando la dependencia exclusiva de proveedores propietarios.

1.3 Roles de los participantes

En este trabajo, cada miembro del equipo asume un rol específico para garantizar una visión integral del tema:

- **Proponente (Julian Carrión Tovar):** Encargado de argumentar las ventajas y la utilidad de integrar GenAI en sistemas distribuidos.
- **Crítico (Minerva Cebrián Marín):** Responsable de identificar y analizar los inconvenientes y riesgos asociados con la implementación de GenAI en entornos distribuidos.
- **Proporcionador de ejemplos (David Bacas Posadas):** Encargado de recopilar y presentar casos de uso reales que demuestren la aplicación práctica de las soluciones propuestas.
- **Preguntador (Miguel Ángel Luque Gómez):** Responsable de formular preguntas técnicas y éticas que surjan del estudio, fomentando el debate y la reflexión en el seminario académico.
- **Resumidor (Pablo Hernández Ibáñez):** Encargado de sintetizar los hallazgos del trabajo y presentar una conclusión clara y concisa que refleje las principales ideas discutidas.

2 Utilidad y Ventajas (Rol: Proponente)

Como hemos mencionado anteriormente, la integración de la IA Generativa en sistemas distribuidos no es un simple capricho técnico, sino una necesidad imperativa para hacer que esta tecnología sea verdaderamente útil, escalable y segura en aplicaciones reales. A continuación, se detallan las principales ventajas prácticas de esta convergencia:

- **Escalabilidad horizontal extrema:** Si un problema es demasiado grande, el sistema permite añadir más recursos en red para superarlo. Esto resulta evidente al observar modelos masivos como GPT-4, los cuales son tan gigantescos que no caben en la memoria de un solo dispositivo, pero pueden ejecutarse sin problema al conectar miles de tarjetas gráficas para que trabajen coordinadas, usando técnicas como el data parallelism, como un único gran cerebro.

- **Eficiencia y optimización de recursos (PagedAttention):** Las arquitecturas modernas gestionan la memoria de forma dinámica para no desperdiciar capacidad. Podemos imaginar una situación donde nuestra universidad lanza un tutor virtual y miles de estudiantes se conectan simultáneamente en época de exámenes; gracias a esta compartición inteligente de la memoria caché, los servidores no colapsan y todos los alumnos reciben su respuesta al instante.

- **Privacidad y soberanía de datos (Aprendizaje Federado):** Es posible entrenar modelos de IA sin que la información confidencial abandone su lugar de origen. Un escenario real de gran utilidad sería el de varios hospitales andaluces colaborando para entrenar un modelo médico; en lugar de enviar historiales a una base de datos central, la IA viaja a cada centro, aprende localmente y solo comparte el conocimiento matemático resultante, manteniendo la privacidad intacta.

- **Transparencia de acceso y ubicación:** El sistema distribuye la carga de trabajo ocultando toda la complejidad subyacente al usuario, permitiendo interactuar con modelos de IA generativa de gran tamaño sin necesidad de conocer su despliegue. Así, un investigador puede ejecutar una petición compleja desde su portátil y la infraestructura repartirá automáticamente el esfuerzo entre múltiples nodos distribuidos, dándole la cómoda ilusión de que todo el procesamiento está ocurriendo mágicamente dentro de su propia máquina.

- **Respuestas inmediatas sin latencia (Edge AI):** Al acercar el cómputo al extremo de la red, además de mejorar la privacidad de los datos al evitar su envío a servidores remotos, se llegan a minimizar o incluso eliminar los tiempos de espera asociados a la transmisión de datos hacia la nube. Resulta fácil comprender la importancia de esto si pensamos en un vehículo autónomo, donde enviar el vídeo de la cámara a un centro de datos lejano para decidir si debe frenar introduciría un retraso letal.

- **Alta disponibilidad y tolerancia a fallos:** El diseño distribuido garantiza que los servicios críticos nunca dejen de funcionar. De este modo, si dependemos de un asistente de IA para gestionar las operaciones de una empresa y un nodo físico sufre un corte eléctrico, el balanceador de carga transfiere inmediatamente el trabajo a otro servidor activo.

- **Desagregación estratégica de tareas:** Este enfoque permite asignar cada fase computacional al hardware que mejor sepa realizarla. Para visualizarlo, cuando un usuario pide, a una IA generativa, analizar un documento muy extenso, el sistema es capaz de enviar la fase inicial de lectura intensiva a los servidores con mayor fuerza bruta, mientras que la posterior generación del texto se delega a máquinas especializadas en ancho de banda, operando en conjunto como una cadena de montaje altamente rentable.

3 Inconvenientes del uso de esta tecnología (Rol: Crítico)

A pesar de las promesas de eficiencia y escalabilidad, la implementación de IA Generativa en sistemas distribuidos introduce desafíos críticos que pueden comprometer la viabilidad del sistema:

- **El límite de la latencia y el ancho de banda:** Aunque se promociona la inmediatez, el rendimiento está intrínsecamente limitado por el retardo de la red. En modelos masivos, la necesidad de sincronizar billones de parámetros entre miles de GPUs genera un overhead de comunicación; si la latencia de red es alta, añadir más recursos (escalabilidad horizontal) puede incluso ralentizar el sistema en lugar de acelerarlo.

- **Desafíos de Integridad y el "Nodo Malicioso":** En entornos como el Aprendizaje Federado o sistemas abiertos, la seguridad es una preocupación mayor. Según el modelo de seguridad de Coulouris , un enemigo puede inyectar mensajes espurios. En IA, esto se traduce en ataques de envenenamiento de datos, donde un solo nodo corrupto puede enviar actualizaciones matemáticas falsas que degraden la precisión de todo el modelo global sin ser detectado.

- **Complejidad y el Teorema CAP:** La integración de IA debe lidiar con el conflicto entre Consistencia y Disponibilidad. Para que la IA funcione como un único cerebro, todos los nodos deben estar perfectamente sincronizados (Consistencia), pero mantener esto en un sistema a gran escala ante posibles fallos de red es matemáticamente imposible de lograr de forma perfecta simultáneamente, según el Teorema CAP.

- **Heterogeneidad:** Los sistemas distribuidos reales no son uniformes. Coulouris destaca la heterogeneidad como un desafío; en una red de hospitales o centros educativos, si un solo servidor es más antiguo o lento, puede retrasar el entrenamiento o la respuesta de todo el sistema distribuido, anulando las ventajas de la computación paralela.

- **Dificultad en la Transparencia de Fallos:** Aunque se busca la ilusión de que todo ocurre en una sola máquina, ocultar la complejidad tiene un precio. Cuando un componente falla, diagnosticar si el error proviene del hardware de lectura, de la red o del nodo de generación es extremadamente complejo, aumentando los costes de mantenimiento operativo.

4 Ejemplos de uso reales (Rol: Proporcionador de ejemplos)

Para ejemplificar como se está utilizando la IA Generativa en sistemas distribuidos en la actualidad, podemos destacar varios casos reales que ilustran su impacto en diferentes sectores:

– **T-Systems:** Cabe matizar la empresa objeto del estudio. Esta plataforma europea permite a industrias reguladas desplegar LLMs de forma segura. Ejemplos incluyen asistentes de notas para el sector público que automatizan la creación de documentos administrativos y asistentes inteligentes en el sector salud para organizar información clínica en tiempo real.

  Esta empresa pone en manifiesto alguna de las ventajas que hemos mencionado anteriormente, por ejemplo el **Aprendizaje Federado**, permitiendo a distintas entidades colaborar sin compartir datos sensibles.

– **Arquitecturas RAG:** Muchas empresas integran LLMs con sus propias bases de conocimiento distribuidas. Esto permite a los chatbots empresariales consultar documentos internos de forma segura, ofreciendo al empleado una **transparencia de acceso total**.

– **Orquestación con Kubernetes y Ray:** En el ámbito del desarrollo, se utilizan pilas de software abierto como estas para el cómputo distribuido. Esto permite ejecutar tareas complejas de procesamiento de vídeo y razonamiento a gran escala de forma eficiente.

  Con esto ejemplificamos la **escalabilidad horizontal extrema**, añadiendo nodos bajo demanda para manejar picos de trabajo sin necesidad de reescribir el código de IA.

– **Amazon Bedrock y Google Cloud:** Estos servicios ofrecen acceso a modelos fundacionales mediante APIs distribuidas, permitiendo a las empresas crear avatares de ventas virtuales o sistemas de automatización de flujos de trabajo sin gestionar la infraestructura física.

  Estas garantizan **alta disponibilidad y transparencia de ubicación**, permitiendo que las empresas automaticen flujos de trabajo globales sin tener que gestionar ni una sola GPU física

– **Despliegue de Tutores Virtuales con vLLM:** Ejemplificando **PagedAttention**, las instituciones educativas pueden servir modelos LLM a miles de alumnos concurrentes. La tecnología de vLLM permite la optimización del hardware disponible.

5 Dudas surgidas del estudio (Rol: Preguntador)

Durante la realización de este trabajo, han surgido varias preguntas técnicas y éticas que consideramos relevantes para el debate en el seminario académico:

– En un sistema distribuido formado por distintos nodos, si un nodo cae, perdemos parte de la memoria compartida. **¿Creen ustedes que deberiamos priorizar la replicación de la memoria de los nodos para evitar esta**

pérdida de memoria, o es preferible asumir la pérdida de memoria y simplemente redirigir las peticiones a otros nodos activos?

- En un escenario donde tenemos hardware de diferentes generaciones, **¿es preferible desechar las GPUs antiguas para mantener una velocidad homogénea, o diseñar estrategias complejas de balanceo de carga para aprovechar todos los recursos?"**

- Conociendo los costes y las necesidades tecnológicas que requieren estos sistemas distribuidos, **¿Creéis que esta arquitectura descentralizada es viable y rentable para una PYME, o es un 'lujo' técnico reservado únicamente para gigantes tecnológicos?**

- En el Aprendizaje Federado, vendemos la idea de que los datos privados nunca salen de su origen. Pero sabiendo que existen ataques de inversión (donde se busca reconstruir datos sensibles usados para entrenar el modelo), **¿hasta qué punto podemos garantizar una verdadera soberanía del dato y como se consigue esto?**

6 Conclusión (Rol: Resumidor)

La integración de la IA Generativa en sistemas distribuidos es ya una necesidad arquitectónica para superar el "muro de la memoria". Esta descentralización permite alcanzar una escalabilidad horizontal extrema y optimizar recursos mediante técnicas como PagedAttention y el Aprendizaje Federado, garantizando la privacidad y viabilidad de modelos masivos que serían inmanejables en dispositivos únicos.

Sin embargo, este enfoque traslada el cuello de botella del procesamiento a la red, enfrentando desafíos críticos de latencia, heterogeneidad de nodos y las limitaciones del Teorema CAP. En definitiva, el éxito de la GenAI no dependerá solo del modelo, sino de la creación de infraestructuras distribuidas que logren armonizar la eficiencia técnica con la seguridad y la soberanía del dato.

7 Auto-evaluaciones

7.1 Auto-evaluación general

Como equipo, consideramos que hemos sido capaces de profundizar bastante en el tópico sobre la integración de la IA Generativa en sistemas distribuidos, adquiriendo conocimientos sobre escalabilidad, Aprendizaje Federado, los retos del Teorema CAP, entre otras. Esta investigación nos ha llevado a analizar tanto ventajas técnicas como riesgos prácticos y éticos, además de investigar sobre casos reales de la industria, consolidando así nuestra capacidad de síntesis y colaboración para comprender y proyectar el tema propuesto de manera clara y rigurosa.

7.2 Auto-evaluación Proponente (Julian Carrión Tovar)

Como responsable de defender la utilidad y las ventajas de integrar la IA Generativa en sistemas distribuidos, mi labor ha consistido en demostrar que esta convergencia no es un capricho técnico, sino una necesidad para lograr modelos verdaderamente escalables y seguros. He defendido la integración de la GenAI no como un capricho técnico, sino como una necesidad para lograr modelos escalables y seguros. He analizado cómo el paralelismo de datos y la escalabilidad horizontal habilitan modelos masivos, proponiendo el uso de *PagedAttention* y el Aprendizaje Federado para optimizar recursos y proteger la privacidad en sectores críticos. Mi meta es demostrar que el diseño distribuido es la única vía para que la IA sea eficiente, resiliente y transparente para el usuario final.

7.3 Auto-evaluación Crítico (Minerva Cebrián Marín)

Como responsable de estudiar los problemas y los riesgos que trae la opción de integrar la IA generativa en sistemas distribuidos, comparo las expectativas de mayor eficiencia con las realidades, tanto técnicas como teóricas. Considero conceptos como el Teorema de CAP y los modelos de seguridad de Coulouris, para hallar vulnerabilidades significativas, como las implicaciones de la latencia en la sobrecarga de comunicación o los envenenamientos de datos en un entorno federado. Mi objetivo es que al analizar estos riesgos, se tengan en cuenta a la hora de la implementación, para así evitarlos o abordarlos de manera efectiva.

7.4 Auto-evaluación Proporcionador de ejemplos (David Bacas Posadas)

Como responsable de analizar los casos de uso reales y las ventajas competitivas de la IA Generativa en arquitecturas distribuidas, mi labor consistió en demostrar cómo la teoría se traduce en soluciones tangibles y seguras. He examinado casos como el Aprendizaje Federado en T-Systems y arquitecturas RAG para probar que el uso de orquestadores (*Kubernetes/Ray*) y técnicas de optimización democratizan el acceso a modelos masivos. Mi enfoque confirma que los sistemas distribuidos son el motor esencial para que la IA sea hoy una herramienta escalable, privada y útil en sectores como la salud, la educación y la industria tecnológica.

7.5 Auto-evaluación Preguntador (Miguel Ángel Luque Gómez)

Como encargado de diseñar las preguntas, he tenido que analizar críticamente los retos que supone la integración de la IA Generativa en arquitecturas de sistemas distribuidos. Mi labor principal ha consistido en identificar los dilemas técnicos y éticos más relevantes de nuestra investigación. Considero que las cuestiones planteadas van más allá de la simple teoría, fomentando un debate crítico que permitirá a cualquier asistente del seminario comprender las implicaciones reales, ventajas e inconvenientes de desplegar estos modelos masivos en el mundo real."

7.6 Auto-evaluación Resumidor (Pablo Hernández Ibáñez)

Como responsable de la preparación del documento final, mi labor ha consistido en consolidar toda la información aportada por mis compañeros,asegurando que cada sección completase un resultado global coherente y comprensible. Me encargué de implementar las distintas secciones así como de redactar una conclusión final, integrando los conocimientos sobre IA Generativa en sistemas distribuidos y reflejando ventajas y retos de sus aplicaciones.

Considero que todos los integrantes han participado de manera activa y que el intercambio de conocimientos fue constante; siendo mi labor la de canalizar estas aportaciones y presentarlas de manera clara, transmitiendo una versión completa sobre el tópico y el Aprendizaje Federado, mostrando cómo permiten escalar, proteger y optimizar los modelos de IA en entornos reales.

References

1. Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2011). *Distributed systems: Concepts and design* (5th ed.). Addison-Wesley.
2. Kwon, W., et al. (2025). *vLLM: An efficient inference engine for large language models* (Technical Report No. UCB/EECS-2025-192). University of California, Berkeley.
3. Author, F. (2025). Serving LLM in distributed GPU cluster with fine-grain pipeline constraints. *IEEE Transactions on Services Computing, 18*(5). https://www.computer.org/csdl/journal/sc/2025/05/11143904/29yj42wwNwY
4. Red Hat Developer. (2025). Why vLLM is the best choice for AI inference today. https://developers.redhat.com/articles/2025/10/30/why-vllm-best-choice-ai-inference-today
5. Meta Intelligence. (2025). Private LLM deployment guide: Llama + vLLM + TensorRT—Self-hosted architecture. https://www.meta-intelligence.tech/en/insight-llm-deployment
6. Author, A.-B. (2025). Federated attention: A distributed paradigm for collaborative LLM inference over edge networks. ResearchGate. https://www.researchgate.net/publication/397279786
7. Solo.io. (2025). Deep dive into llm-d and distributed inference. https://www.solo.io/blog/deep-dive-into-llm-d-and-distributed-inference
8. Author, F. (2025). Performance modeling and workload analysis of distributed large language model training and inference. ResearchGate. https://www.researchgate.net/publication/387919319
9. T-Systems. (2025). AI foundation services. https://www.t-systems.com/de/en/artificial-intelligence/solutions/ai-foundation-services
10. Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2012). *Distributed systems: Concepts and design* (5th ed.). Pearson Education.
11. Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News, 33*(2), 51–59.