Carrera: Ingeniería en Sistemas Computacionales

Materia: Arquitectura de soluciones en la nube

Docente: Barradas Hernández Karen Andrea

Unidad: 3 – Bases de Datos en la Nube

Actividad: Configuración y Optimización de Bases de Datos

Estudiante:

227O01221 URIEL GARCÍA GARCÍA

Semestre: 7° – Grupo: A - Modalidad: Sabatino

Lugar y Fecha: Xalapa, Ver a 23/10/2025

EJERCICIO

Ejercicio centrado en un problema diferente: la contención de cargas de trabajo (analítica vs. transaccional).

Ejercicio Práctico: Optimización de Base de Datos "GestorPro"

Objetivo del Ejercicio: Diagnosticar un problema de rendimiento causado por la interferencia entre diferentes tipos de carga de trabajo (OLTP vs. OLAP) y proponer un plan de optimización que priorice el aislamiento de cargas y la eficiencia de costos.

Escenario del Problema

Eres el arquitecto de nube de "GestorPro", una popular herramienta SaaS (Software as a Service) de gestión de proyectos. La aplicación funciona 24/7 para usuarios de todo el mundo.

Recientemente, la empresa lanzó una nueva función de "Dashboard de Analítica" para los gerentes. Esta función permite a los gerentes ejecutar reportes complejos sobre la productividad del equipo, el estado de los proyectos y los costos.

Desde el lanzamiento, el equipo de soporte está inundado de quejas: la aplicación principal (crear tareas, mover tarjetas, añadir comentarios) se vuelve extremadamente lenta, casi inutilizable, durante las últimas horas de la tarde en EE.
UU. (aprox. 4:00 PM - 6:00 PM), que es cuando los gerentes ejecutan sus reportes de "fin del día".

El equipo de infraestructura intentó solucionar esto escalando verticalmente la base de datos a una instancia muy grande y costosa, pero el problema de lentitud persiste.

Configuración Actual:

- **Instancia de BD:** db.m5.4xlarge (Optimizada para Memoria: 16 vCPU, 64 GB RAM)
- **Tipo de BD:** MySQL Gestionada
- **Arquitectura:** Una única instancia de base de datos que maneja **todo**: las transacciones de la aplicación (OLTP) y los reportes de analítica (OLAP).

D Panel de Monitoreo (Datos de las últimas 2 semanas)

Observas las métricas y notas un patrón claro:

Métrica	Promedio (8 AM - 4 PM)	Pico (4 PM - 6 PM)	Notas
Uso de CPU	60%	98%	La CPU se satura por las tardes.
Uso de Memoria (RAM)	70%	75%	La memoria parece estar bien (gracias a la instancia m5).
Operaciones de Lectura (IOPS)	3,000 IOPS	25,000 IOPS	Los dashboards de BI leen tablas enteras.
Operaciones de Escritura (IOPS)	2,500 IOPS	2,500 IOPS	El uso normal de la app (crear tareas) es estable.
Latencia de Consultas (ESCRITURA)	50 ms	1,800 ms	Guardar una tarea tarda casi 2 segundos.

Tarea para los Estudiantes

1.- Diagnóstico del Problema:

- ¿Por qué la estrategia de Escalado Vertical (¿a una instancia db? m5.4xlarge) no solucionó el problema, a pesar de que la memoria (75%) no está saturada? El escalado vertical aumenta la capacidad general de la instancia (más vCPU y RAM), lo cual ayuda en escenarios de carga uniforme, pero no resuelve problemas de *contención de recursos* entre cargas de trabajo diferentes. En este caso, la memoria no es el cuello de botella (está en 75%, con margen), pero las consultas OLAP (reportes de analítica) generan un "ruido" intensivo en lecturas que bloquea o compite por recursos compartidos (como locks en tablas, buffers de I/O o el planificador de consultas del motor de MySQL). Esto afecta desproporcionadamente las operaciones OLTP (escrituras transaccionales), independientemente del tamaño de la instancia. Simplemente "echar más hardware" no separa las cargas conflictivas; solo diluye temporalmente el problema hasta que el volumen de OLAP crezca.

- ¿Cuál es el verdadero cuello de botella? (No es solo la CPU). Relaciona las métricas de Lectura y Escritura. El cuello de botella principal es la *contención entre cargas OLTP y OLAP*, donde las lecturas intensivas (OLAP) interfieren con las escrituras (OLTP). Las métricas lo evidencian: las lecturas IOPS se disparan a 25,000 durante el pico (debido a los dashboards que escanean tablas enteras para reportes), mientras que las escrituras IOPS permanecen estables en 2,500 (uso normal de la app). Sin embargo, la latencia de escrituras salta de 50 ms a 1,800 ms, lo que indica que las consultas OLAP están causando *bloqueos de recursos* (e.g., locks en filas/tablas durante lecturas pesadas) o saturando el subsistema de I/O/buffer pool, retrasando las transacciones OLTP críticas. La CPU al 98% agrava esto, pero es un síntoma; el conflicto de workloads es la raíz, ya que una sola instancia no puede priorizar o aislar eficientemente estas cargas heterogéneas.

2. Propuesta de Plan de Optimización:

Mi plan de acción consta de 3 pasos secuenciales, priorizando el aislamiento de cargas para resolver la latencia de escrituras (1,800 ms) y reduciendo costos al optimizar la instancia principal. Se basa en las estrategias de Escalado (Horizontal

para lecturas), Dimensionamiento Óptimo (Right-sizing para la maestra) y Monitoreo/Automatización (para queries y alertas).

- Paso 1: Implementar Réplicas de Lectura Dedicadas para Cargas OLAP (Escalado Horizontal) Descripción: Configurar 2-3 réplicas de lectura (read replicas) en instancias más pequeñas (e.g., db.m5.large: 2 vCPU, 8 GB RAM) y redirigir todas las consultas de analítica (dashboards y reportes) a estas réplicas mediante un enrutador de consultas o modificaciones en la aplicación (e.g., usando drivers de MySQL que separen OLTP de OLAP). La instancia maestra manejará solo escrituras y lecturas ligeras de la app principal. Estrategia Principal: Escalado Horizontal. Justificación: Esto aislará las lecturas intensivas (25,000 IOPS) de las escrituras (2,500 IOPS), eliminando la contención que causa la latencia de 1,800 ms en escrituras. Las réplicas absorberán el "ruido" OLAP sin afectar la maestra, resolviendo el cuello de botella principal y mejorando el rendimiento OLTP en picos.

- Paso 2: Realizar Right-Sizing en la Instancia Maestra (Dimensionamiento Óptimo) Descripción: Una vez implementadas las réplicas, downgradear la instancia maestra de db.m5.4xlarge (16 vCPU, 64 GB RAM) a una db.m5.2xlarge (8 vCPU, 32 GB RAM), enfocada en workloads transaccionales estables. Monitorear por 1 semana post-cambio para validar. Estrategia Principal: Dimensionamiento Óptimo (Right-sizing). Justificación: Con las lecturas OLAP offloadadas, la maestra solo manejará escrituras estables (CPU promedio 60%, memoria 70%), haciendo innecesarios los 16 vCPU extras. Esto reduce la latencia residual en escrituras al liberar recursos no saturados y corta costos en ~50% (la instancia maestra es el gasto 24/7 más alto), abordando la contención sin sobredimensionamiento.

- Paso 3: Configurar Monitoreo de Queries y Alertas Automáticas (Monitoreo y Automatización) Descripción: Activar herramientas nativas de MySQL gestionada (e.g., Performance Schema o CloudWatch en AWS) para monitorear queries por tipo (OLTP vs. OLAP), con alertas que notifiquen si las lecturas OLAP exceden el 20% del tráfico total. Automatizar redirecciones o pausas en dashboards si detectan contención (e.g., vía Lambda functions). Estrategia Principal: Monitoreo y Automatización. Justificación: Identificará queries OLAP ineficientes que aún causen

latencia en escrituras (e.g., scans completos de tablas), permitiendo optimizaciones como índices o materialización de vistas. La automatización previene picos futuros, manteniendo la latencia <100 ms y evitando escalados reactivos que eleven costos.

3. Pregunta de Monitoreo Preventivo

Para evitar que el nuevo "Dashboard de Finanzas" repita el problema de contención, implementaría una estrategia de Monitoreo Predictivo y Automatización de Workload Isolation antes del lanzamiento:

- Monitoreo: Configurar un dashboard proactivo (e.g., usando Prometheus o el monitoring de la nube) que clasifique queries en tiempo real por patrón (OLTP: escrituras rápidas; OLAP: agregaciones/joins complejos) y mida su impacto en métricas clave (latencia de escrituras, locks activos). Establecer umbrales predictivos basados en pruebas de carga simuladas (e.g., alertar si queries OLAP proyectadas >15,000 IOPS).

- Automatización: Desarrollar reglas de enrutamiento dinámico (e.g., con ProxySQL o aplicación-level routing) que automaticen la desviación de nuevas queries OLAP a réplicas dedicadas o un clúster separado de analítica (e.g., un data warehouse como Amazon Redshift para finanzas). Incluir un "sandbox" de testing pre-lanzamiento: ejecutar el dashboard en un entorno staging y simular tráfico mixto para validar aislamiento, con auto-escalado de réplicas si la contención supera el 10%. Esto asegura que el lanzamiento no impacte OLTP, manteniendo costos controlados mediante auto-downsizing post-pico.