Enfoque de informática de alto rendimiento

Marco de Buena Arquitectura de AWS

Diciembre de 2019

This paper has been archived.

The latest version is now available at:

https://docs.aws.amazon.com/es_es/wellarchitected/latest/high-performance-computing-lens/welcome.html

Avisos

Los clientes son responsables de realizar sus propias evaluaciones de la información contenida en este documento. Este documento: (a) solo tiene fines informativos, (b) representa las prácticas y las ofertas de productos vigentes de AWS, que están sujetas a cambios sin previo aviso, y (c) no crea ningún compromiso ni garantía de AWS y sus empresas afiliadas, proveedores o licenciantes. Los productos o servicios de AWS se proporcionan “tal cual”, sin garantías, representaciones ni condiciones de ningún tipo, ya sean explícitas o implícitas. Las responsabilidades y obligaciones de AWS con respecto a sus clientes se controlan mediante los acuerdos de AWS; este documento no forma parte de ningún acuerdo entre AWS y sus clientes ni lo modifica.

Contenido

Resumen

En este documento se describe el **enfoque de informática de alto rendimiento (HPC)** para el Marco de Buena Arquitectura de AWS. El documento abarca escenarios comunes de HPC e identifica elementos clave para garantizar que las cargas de trabajo se diseñen de acuerdo con las prácticas recomendadas.

Introducción

El Marco de Buena Arquitectura de AWS lo ayuda a comprender las ventajas y desventajas de las decisiones que toma cuando crea sistemas en AWS.[1] Utilice el marco para aprender sobre las prácticas recomendadas de arquitectura para diseñar y operar sistemas confiables, seguros, eficientes y rentables en la nube. El marco ofrece una forma de medir sus arquitecturas de forma constante en función de las prácticas recomendadas e identificar áreas de mejora. Creemos que contar con sistemas de buena arquitectura aumenta en gran medida la probabilidad de éxito empresarial.

En este "enfoque" nos centramos en cómo diseñar, implementar y crear sus **cargas de trabajo de informática de alto rendimiento (HPC)** en la nube de AWS. Las cargas de trabajo de HPC se ejecutan excepcionalmente bien en la nube. La fluctuación natural y el estallido característico de las cargas de trabajo de HPC las hace apropiadas para la infraestructura de la nube de pago por uso. La capacidad de ajustar los recursos en la nube y crear arquitecturas nativas de la nube acelera naturalmente el cambio de las cargas de trabajo de HPC.

Por cuestiones de brevedad, solo hemos cubierto detalles del marco de buena arquitectura que son específicos de las cargas de trabajo de HPC. Aconsejamos que considere las prácticas recomendadas y preguntas desde el documento técnico del Marco de Buena Arquitectura de AWS cuando[2] diseñe su arquitectura.

Este documento está destinado a aquellos que ocupan puestos en tecnología, como los directores de tecnología (CTO), arquitectos, desarrolladores y miembros del equipo de operaciones. Después de leer este documento, comprenderá mejor las prácticas recomendadas y estrategias de AWS para utilizar cuando diseñe y opere HPC en un entorno de nube.

Definiciones

El Marco de Buena Arquitectura de AWS se basa en cinco pilares: excelencia operativa, seguridad, confiabilidad, eficiencia del rendimiento y optimización de costes. Cuando diseña soluciones, compensa entre pilares basados en el contexto de su negocio. Estas decisiones de negocios pueden impulsar sus prioridades de diseño. Puede reducir el coste a expensas de la confiabilidad en el desarrollo de entornos o, para soluciones de misión crítica, puede optimizar la confiabilidad con costes aumentados. La seguridad y la excelencia operativa generalmente no se negocian contra otros pilares.

Mediante este informe, hacemos una distinción crucial entre cargas de trabajo levemente acopladas, a veces denominadas informática de alto rendimiento (HTC) en la comunidad, y cargas de trabajo estrechamente acopladas. También cubrimos diseños basados en servidor o sin servidor. Consulte a la sección de escenarios para una discusión detallada sobre estas distinciones.

Parte del vocabulario de la nube de AWS puede diferir de la terminología común de HPC. Por ejemplo, los usuarios de HPC pueden referirse a un servidor como un “nodo”, mientras que AWS se refiere a un servidor virtual como una “instancia”. Cuando los usuarios de HPC comúnmente hablan de “trabajos”, AWS se refiere a ellos como “cargas de trabajo”.

La documentación de AWS utiliza el término “CPU virtual” como sinónimo de “subproceso” o “hyperthread” (o la *mitad* de un núcleo físico). No se pierda este factor de 2 cuando cuantifique el rendimiento o coste de una aplicación de HPC en AWS.

Los **grupos de ubicación en clúster** son un método de AWS para agrupar sus instancias de informática para aplicaciones con los requisitos de red más altos. Un grupo de ubicación no es un elemento de hardware físico. Simplemente es una regla lógica que mantiene todos los nodos dentro de un radio de baja latencia de la red.

La infraestructura de la nube de AWS está compuesta de **regiones** y **zonas de disponibilidad**. Una región es una ubicación física en el mundo donde disponemos de varias zonas de disponibilidad. Las zonas de disponibilidad constan de uno o varios centros de datos discretos, cada uno de ellos con alimentación, redes y conectividad redundantes, que se alojan en instalaciones independientes. Según las características de su carga de trabajo de HPC, es posible que desee que su clúster abarque zonas de disponibilidad (aumenta la confiabilidad) o permanezca dentro de una sola zona de disponibilidad (enfatiza la baja latencia).

Principios generales de diseño

En los entornos informáticos tradicionales, las decisiones sobre arquitectura se implementan a menudo como eventos estáticos, únicos, a veces sin actualizaciones importantes de

software o hardware durante la vida útil de un sistema informático. A medida que un proyecto y su contexto evolucionan, estas decisiones iniciales pueden dificultar la capacidad del sistema para cumplir con los requisitos comerciales cambiantes.

Es diferente en la nube. Una infraestructura en la nube puede crecer a medida que crece el proyecto, lo que permite una capacidad de optimización continua. En la nube, la capacidad de automatizar y probar bajo demanda reduce el riesgo de impacto de los cambios en el diseño de la infraestructura. Esto permite que los sistemas evolucionen con el tiempo para que los proyectos puedan aprovechar las innovaciones como una práctica estándar.

El marco de buena arquitectura propone un conjunto de principios generales de diseño para facilitar un buen diseño en la nube con informática de alto rendimiento:

- **Arquitecturas dinámicas**: evitar arquitecturas congeladas y estáticas, y las estimaciones de costes que utilizan un modelo de estado estable. Su arquitectura debe ser dinámica: con crecimiento y reducción para coincidir con sus demandas de HPC a través del tiempo. Coincidir con el diseño de su arquitectura y análisis de costes explícitamente para los ciclos naturales de las actividades de HPC. Por ejemplo, un periodo de intensos esfuerzos de simulación puede estar seguido por una reducción en la demanda cuando el trabajo pasa de la fase de diseño al laboratorio. O una fase de acumulación de datos larga y progresiva puede estar seguida por un análisis a gran escala y una fase de reducción de datos. A diferencia de muchos centros de supercomputación tradicional, la nube de AWS lo ayuda a evitar largas colas, largas aplicaciones de cuotas y restricciones en la personalización e instalación del software. Muchos esfuerzos de HPC son intrínsecamente ampliables y coinciden con los paradigmas de la nube de elasticidad y pago por uso. La elasticidad y el modelo de pago por uso de AWS eliminan la dolorosa elección entre sistemas con exceso de suscripción (espera en colas) o sistemas inactivos (dinero desperdiciado). Entornos, como los clústeres de informática, pueden tener el “tamaño adecuado” para una necesidad determinada en cualquier momento.
- **Ajustar el modelo de adquisición a la carga de trabajo**: AWS pone a disposición una variedad de modelos de adquisición de cómputo para los diversos patrones de uso de HPC. La selección del modelo correcto garantiza que solo pague por lo que necesita. Por ejemplo, un instituto de investigación puede ejecutar la misma aplicación de pronóstico meteorológico de diferentes maneras:
    - Un proyecto de investigación académica estudia el papel de una variable meteorológica con un gran número barridos y conjuntos de parámetros. Estas simulaciones no son urgentes y el coste es una preocupación principal. Son una excelente combinación para las instancias de spot de Amazon EC2. Las instancias de spot permiten tomar ventaja de la capacidad sin utilizar de Amazon EC2 y

están disponibles hasta con un 90 % de descuento comparado con los precios bajo demanda.

  - Durante la temporada de incendios forestales, los pronósticos de viento locales actualizados garantizan la seguridad de los bomberos. Cada minuto de demora en las simulaciones disminuye su oportunidad de evacuación segura. Para estas simulaciones, se deben utilizar las instancias bajo demanda para permitir un amplio análisis y garantizar que los resultados se obtengan sin interrupción.
  - Todas las mañanas, los pronósticos meteorológicos emiten para transmisiones de televisión de la tarde. Las instancias reservadas programadas pueden utilizarse para asegurar que la capacidad necesaria esté disponible todos los días en el momento adecuado. El uso de este modelo de precios ofrece un descuento comparado con las instancias bajo demanda.

- **Comenzar desde los datos**: antes de comenzar a diseñar su arquitectura, debe tener una imagen clara de los datos. Considere el origen, el tamaño, la velocidad y las actualizaciones de los datos. Una optimización holística del rendimiento y coste se centra en calcular e incluir las consideraciones de los datos. AWS tiene una fuerte oferta de datos y servicios relacionados, incluidos la visualización de datos, los cuales le permiten extraer el máximo valor de sus datos.
- **Automatizar para simplificar la experimentación arquitectónica**: la automatización mediante códigos permite crear y replicar sus sistemas a bajo costo y evitar los gastos del esfuerzo manual. Puede rastrear cambios en su código, auditar su impacto y volver a las versiones anteriores cuando sea necesario. La habilidad de experimentar fácilmente con la infraestructura permite optimizar la arquitectura para el rendimiento y el coste. AWS ofrece herramientas, como AWS ParallelCluster, que lo ayudan a comenzar a tratar su infraestructura de nube de HPC como un código.
- **Habilitar la colaboración**: el trabajo de HPC a menudo ocurre en un contexto de colaboración, que a veces abarca muchos países de todo el mundo. Más allá de la colaboración inmediata, los métodos y resultados a menudo se comparten con la más amplia comunidad científica y de HPC. Es importante considerar con antelación qué herramientas, código y datos se pueden compartir y con quién. Los métodos de entrega deben ser parte de este proceso de diseño. Por ejemplo, los flujos de trabajo pueden compartirse de muchas maneras en AWS: puede utilizar Imágenes de Amazon Machine (AMI), instantáneas de Amazon Elastic Block Store (Amazon EBS), buckets de Amazon Simple Storage Service (Amazon S3), plantillas de AWS CloudFormation, archivos de configuración de AWS ParallelCluster, productos de AWS Marketplace y scripts. Saque el máximo provecho de las características de colaboración y seguridad

de AWS, que hacen de AWS un entorno excelente para que usted y sus colaboradores puedan resolver sus problemas de HPC. Esto ayuda a que sus soluciones informáticas y conjuntos de datos alcancen un mayor impacto cuando se comparten de forma segura dentro de un grupo selecto o públicamente con la comunidad en general.

- **Usar diseños nativos en la nube**: por lo general, no es necesario ni óptimo replicar su entorno en las instalaciones cuando migra las cargas de trabajo a AWS. La amplitud y profundidad de los servicios de AWS permite que las cargas de trabajo de HPC se ejecuten de nuevas formas mediante nuevos patrones de diseño y soluciones nativas en la nube. Por ejemplo, cada usuario o grupo puede utilizar un clúster separado, el cual puede escalar independientemente según la carga. Los usuarios pueden confiar en un servicio administrado, como AWS Batch, o informática sin servidor, como AWS Lambda, para administrar la infraestructura subyacente. Considere no utilizar un programador de clúster tradicional y en su lugar usar un programador solo si la carga de trabajo lo requiere. En la nube, los clústeres de HPC no requieren permanencia y pueden ser recursos efímeros. Cuando automatiza la implementación de su clúster, puede terminar un clúster y lanzar otro nuevo rápidamente con los mismos o diferentes parámetros. Este método crea entornos según sea necesario.
- **Probar cargas de trabajo de la vida real**: la única forma de saber cómo funcionará su carga de trabajo de producción en la nube es probarla en la nube. La mayoría de las aplicaciones de HPC son complejas, y sus patrones de memoria, CPU y red a menudo no pueden reducirse a una simple prueba. Además, los requisitos de la aplicación para la infraestructura varían en función de los solucionadores de aplicaciones (métodos o algoritmos matemáticos) que utilizan sus modelos, el tamaño y la complejidad de sus modelos, etc. Por esta razón, los puntos de referencia genéricos no son variables confiables del rendimiento de producción de HPC real. Del mismo modo, tiene poco valor probar una aplicación con un pequeño punto de referencia o “problema con poco interés científico”. Con AWS, solo paga por lo que realmente usa. Por lo tanto, es factible hacer una prueba de concepto realista con sus propios modelos representativos. Una ventaja importante de una plataforma basada en la nube es que se puede realizar una prueba realista a gran escala antes de la migración.
- **Equilibrar el tiempo de resultados y la reducción de costes:** analizar el rendimiento con los parámetros más significativos: el tiempo y el coste. Se debe usar el enfoque en la optimización de costes para cargas de trabajo que no son urgentes. Las instancias de spot suelen ser el método menos costoso para cargas de trabajo que no son urgentes. Por ejemplo, si un investigador cuenta con una gran cantidad de medidas de laboratorio que se deben analizar en algún momento antes de la conferencia del año siguiente, las instancias de spot pueden ayudar a analizar la mayor cantidad posible de

medidas dentro del presupuesto fijo de investigación. Por el contrario, para cargas que no son urgentes, como el modelado de respuesta a emergencias, la optimización de costes puede cambiarse para el rendimiento. Además, se deben elegir el tipo de instancia, el modelo de adquisición y el tamaño del clúster para lograr el tiempo de ejecución más bajo e inmediato. Si se comparan plataformas, es importante tener en cuenta todo el tiempo de búsqueda de una solución, incluidos los aspectos que no son informáticos, como el aprovisionamiento de recursos, los datos transitorios o, en entornos más tradicionales, el tiempo dedicado a las colas de trabajo.

Escenarios

Los casos de HPC suelen ser problemas informáticos complejos que requieren técnicas de procesamiento en paralelo. Una infraestructura de HPC de buena arquitectura puede lograr un rendimiento sostenido por el tiempo que duren los cálculos para respaldar los cálculos. Las cargas de trabajo de HPC abarcan aplicaciones tradicionales, como la genómica, la química informática, el modelado de riesgos financieros, la ingeniería asistida por equipo, la predicción meteorológica y la obtención de imágenes sísmicas, así como las aplicaciones emergentes, tales como el aprendizaje automático, el aprendizaje profundo y la conducción autónoma. Aun así, las redes tradicionales o los clústeres de HPC que son compatibles con estos cálculos son notablemente similares en arquitectura con atributos de clúster selectos optimizados para la carga de trabajo específica. En AWS, se pueden elegir estratégicamente la red, el tipo de almacenamiento, el tipo de cómputo (instancia) e incluso el método de implementación para optimizar el rendimiento, el coste y el uso de una carga de trabajo particular.

La HPC se divide en dos categorías según el grado de interacción entre los procesos en paralelo que se ejecutan de forma simultánea: cargas de trabajo levemente acopladas y cargas de trabajo estrechamente acopladas. Los casos de HPC levemente acoplada son aquellos en los que los procesos múltiples o en paralelo no interactúan estrechamente entre sí durante toda la simulación. Los casos de HPC estrechamente acoplada son aquellos en los que los procesos en paralelo se ejecutan de forma simultánea e intercambian información con regularidad entre sí en cada iteración o paso de la simulación.

Con las cargas de trabajo levemente acopladas, la finalización de un cálculo o una simulación completos a menudo requiere cientos o millones de procesos en paralelo. Estos procesos se producen en cualquier orden y a cualquier velocidad durante la simulación. Esto ofrece flexibilidad en la infraestructura informática necesaria para las simulaciones levemente acopladas.

Las cargas de trabajo estrechamente acopladas cuentan con procesos que intercambian información con regularidad en cada iteración de la simulación. Por lo general, estas simulaciones estrechamente acopladas se ejecutan en un clúster homogéneo. Si la infraestructura lo permite, la cantidad total de núcleos o procesadores puede variar entre decenas, miles y, en ocasiones, cientos de miles. Las interacciones de los procesos durante la

simulación imponen más exigencias a la infraestructura, como los nodos de computación y la infraestructura de la red.

La infraestructura que se utiliza para ejecutar la gran variedad de aplicaciones levemente y estrechamente acopladas se diferencia por su capacidad de procesar interacciones en los nodos. Existen aspectos fundamentales que se aplican tanto a los escenarios como a las consideraciones específicas de diseño de cada uno de ellos. Tenga en cuenta los siguientes aspectos fundamentales para ambos escenarios cuando seleccione una infraestructura de HPC en AWS:

- **Red:** los requisitos de la red pueden variar entre casos con pocos requisitos, como aplicaciones levemente acopladas con un mínimo de tráfico de comunicación, y aplicaciones estrechamente acopladas y masivamente paralelas que necesitan una red de mayor rendimiento con un gran ancho de banda y baja latencia.
- **Almacenamiento:** los cálculos de HPC utilizan, crean y transfieren datos de maneras exclusivas. La infraestructura de almacenamiento debe ser compatible con estos requisitos durante cada paso del cálculo. Los datos de entrada se almacenan con frecuencia en el inicio y, además, se crean y almacenan más datos en la ejecución. Por otro lado, los datos de salida se transfieren a una ubicación de reserva cuando finaliza la ejecución. Entre los factores a tener en cuenta se incluyen el tamaño de los datos, el tipo de medio, las velocidades de transferencia, el acceso compartido y las propiedades de almacenamiento (por ejemplo, durabilidad y disponibilidad). Resulta útil utilizar un sistema de archivos compartidos entre los nodos. Por ejemplo, mediante el uso de un recurso compartido de sistema de archivos de red (NFS), como Amazon Elastic File System (EFS), o un sistema de archivos de Lustre, como Amazon FSx for Lustre.
- **Informática:** el tipo de instancia de Amazon EC2 define las capacidades del hardware que se disponen para su carga de trabajo de HPC. Entre las capacidades del hardware se incluyen el tipo de procesador, la frecuencia del núcleo, las características del procesador (por ejemplo, las extensiones vectoriales), la relación memoria/núcleo y el rendimiento de la red. En AWS, se considera que una instancia es lo mismo que un nodo de HPC. En este documento técnico, estos términos se utilizan indistintamente.
    - AWS ofrece servicios administrados con la capacidad de acceder a la informática sin la necesidad de elegir el tipo de instancia EC2 subyacente. AWS Lambda y AWS Fargate son servicios informáticos que permiten ejecutar cargas de trabajo sin tener que aprovisionar y administrar los servidores subyacentes.
- **Implementación:** AWS ofrece muchas opciones para implementar cargas de trabajo de HPC. Desde la consola de administración de AWS se pueden lanzar las instancias de manera manual. Para lograr una implementación automatizada, tiene disponible una

variedad de kits de desarrollo de software (SDK) para codificar soluciones de extremo a extremo en diferentes lenguajes de programación. Una opción conocida de la implementación de HPC combina el scripting de bash shell con la interfaz de línea de comandos de AWS (AWS CLI).

- Las plantillas de AWS CloudFormation permiten especificar los clústeres de HPC adaptados a la aplicación que se describen como código para que se puedan iniciar en minutos. AWS ParallelCluster es un software de código abierto que coordina el lanzamiento de un clúster a través de CloudFormation con el software ya instalado (por ejemplo, compiladores y programadores) para lograr una experiencia de clúster tradicional.
- AWS proporciona servicios de implementación administrados para cargas de trabajo basadas en contenedores, como Amazon EC2 Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), AWS Fargate y AWS Batch.
- Las empresas externas disponen de opciones de software adicionales en AWS Marketplace y la red de socios de AWS (APN).

La informática en la nube facilita la experiencia con los componentes de la infraestructura y el diseño de la arquitectura. AWS recomienda encarecidamente evaluar los tipos de instancia, los tipos de volumen de EBS, los métodos de implementación, etc., a fin de buscar el mejor rendimiento al costo más bajo.

Escenarios levemente acoplados

Una carga de trabajo levemente acoplada implica el procesamiento de una gran cantidad de trabajos más pequeños.. Por lo general, el trabajo más pequeño se ejecuta en un nodo, ya sea mediante el consumo de uno o varios procesos con paralelización de memoria compartida (SMP) para la paralelización dentro de ese nodo.

Los procesos en paralelo, o las iteraciones en la simulación, se procesan posteriormente para crear una solución o un descubrimiento a partir de la simulación. Las aplicaciones levemente acopladas se encuentran en muchas áreas, como las simulaciones de Monte Carlo, el procesamiento de imágenes, los análisis genómicos y la automatización de diseño electrónico (EDA).

La pérdida de un nodo o trabajo en una carga de trabajo levemente acoplada no suele retrasar todo el cálculo. El trabajo perdido puede retomarse más tarde u omitirse por completo. Los nodos implicados en el cálculo pueden varias en especificación y potencia.

Una arquitectura adecuada para una carga de trabajo levemente acoplada tiene las siguientes consideraciones:

- **Red:** debido a que los procesos en paralelo no suelen interactuar entre sí, la viabilidad o el rendimiento de las cargas de trabajo no es sensible al ancho de banda ni a las

capacidades de latencia de la red entre instancias. Por lo tanto, no se necesitan los grupos de ubicación en clústeres en este escenario porque debilitan la resistencia sin brindar ganancia en el rendimiento.

- **Almacenamiento:** las cargas de trabajo levemente acopladas varían en cuanto a los requisitos de almacenamiento, y dependen del tamaño del conjunto de datos y del rendimiento deseado para transferir, leer y escribir los datos.
- **Informática:** cada aplicación es diferente, pero por lo general, la relación memoria/cálculo de la aplicación impulsa el tipo de instancia EC2 subyacente. Se optimizan algunas aplicaciones para aprovechar las ventajas de las unidades de procesamiento de gráficos (GPU) o de los aceleradores de la matriz de puertas programables en campo (FPGA) en las instancias EC2.
- **Implementación:** las simulaciones levemente acopladas a menudo se ejecutan en muchos núcleos (a veces millones de núcleos) de informática que se pueden distribuir en las zonas de disponibilidad sin sacrificar el rendimiento. Las simulaciones levemente acopladas se pueden implementar con servicios y soluciones de extremo a extremo, tales como AWS Batch y AWS ParallelCluster, o mediante combinaciones de servicios de AWS, tales como Amazon Simple Queue Service (Amazon SQS), Auto Scaling, AWS Lambda y AWS Step Functions.

Escenarios estrechamente acoplados

Las aplicaciones estrechamente acopladas constan de procesos en paralelo que dependen unos de otros para llevar a cabo el cálculo. A diferencia de un cálculo levemente acoplado, todos los procesos de una simulación estrechamente acoplada se iteran juntos y requieren comunicación entre sí. Una iteración es un paso de una simulación general. Los cálculos estrechamente acoplados dependen de decenas a miles de procesos o núcleos a lo largo de una o millones de iteraciones. Por lo general, la falla de un nodo conduce a la falla de todo el cálculo. Para mitigar el riesgo de que falle todo, se realiza un control a nivel de aplicación con regularidad durante un cálculo para que una simulación se reinicie desde un estado conocido.

Estas simulaciones dependen de una interfaz de paso de mensajes (MPI) para la comunicación entre procesos. El paralelismo de memoria compartida a través de OpenMP se puede usar con una MPI. Los ejemplos de cargas de trabajo estrechamente acopladas de HPC incluyen: dinámica de fluidos computacional, predicción meteorológica y simulación de yacimientos.

Una arquitectura adecuada para una carga de trabajo estrechamente acoplada de HPC tiene las siguientes consideraciones:

- **Red:** los requisitos de la red para los cálculos estrechamente acoplados son exigentes. La comunicación lenta entre nodos ralentiza todo el cálculo. Se necesita el tamaño de

instancia más grande, una mejora en la red y grupos de ubicación en clúster para lograr un rendimiento de red coherente. Estas técnicas minimizan los tiempos de ejecución de la simulación y reducen los costes totales. Las aplicaciones estrechamente acopladas varían en cuanto al tamaño. Un problema de gran tamaño, distribuido en una gran cantidad de procesos o núcleos, generalmente se paraleliza bien. Los casos pequeños, con menores requisitos computacionales totales, son los que más demandan en la red. Algunas instancias de Amazon EC2 utilizan Elastic Fabric Adapter (EFA) como una interfaz de red que permite que se ejecuten aplicaciones que necesitan altos niveles de comunicaciones entre nodos a escala en AWS. La interfaz de hardware de derivación del sistema operativo de EFA mejora el rendimiento de las comunicaciones entre instancias, que es fundamental para escalar aplicaciones estrechamente acopladas.

- **Almacenamiento:** las cargas de trabajo estrechamente acopladas varían en cuanto a los requisitos de almacenamiento, y dependen del tamaño del conjunto de datos y del rendimiento deseado para transferir, leer y escribir los datos. El almacenamiento temporal de datos o el espacio temporal requiere una consideración especial.
- **Informática:** las instancias EC2 se ofrecen en distintas configuraciones con diferentes relaciones de núcleo a memoria. En el caso de las aplicaciones en paralelo, es útil distribuir simulaciones en paralelo con uso intensivo de memoria a través de más nodos de computación para disminuir los requisitos de memoria por núcleo y para dirigirse al tipo de instancia de mejor rendimiento. Las aplicaciones estrechamente acopladas necesita un clúster homogéneo construido a partir de nodos parecidos de computación. Seleccionar el tamaño de instancia más grande minimiza la latencia de la red entre nodos, mientras proporciona el máximo rendimiento de la red cuando se comunica entre nodos.
- **Implementación:** dispone de distintas opciones de implementación. Se puede lograr una automatización de extremo a extremo, así como el lanzamiento de la simulación en un entorno “tradicional” de clúster. La escalabilidad de la nube le permite lanzar cientos de grandes casos de múltiples procesos a la vez, por lo que no hay necesidad de esperar en una cola. Las simulaciones estrechamente acopladas se pueden implementar con soluciones de extremo a extremo, tales como AWS Batch y AWS ParallelCluster, o a través de soluciones basadas en los servicios de AWS, tales como CloudFormation o EC2 Fleet.

Arquitecturas de referencia

Se aplican muchas arquitecturas a las cargas de trabajo levemente y estrechamente acopladas, y es posible que se necesiten realizar leves modificaciones según el escenario. Los

clústeres en las instalaciones tradicionales obligan a un enfoque único para la infraestructura del clúster. Sin embargo, la nube ofrece una amplia gama de posibilidades y permite optimizar el rendimiento y el coste. En la nube, su configuración puede variar desde una experiencia de clúster tradicional con un planificador y un nodo de inicio de sesión, hasta una arquitectura nativa de la nube con las ventajas de la rentabilidad obtenible con soluciones nativas en la nube. A continuación se presentan cinco arquitecturas de referencia:

1. Entorno de clúster tradicional
2. Arquitectura basada en lotes
3. Arquitectura basada en colas
4. Implementación híbrida
5. Flujo de trabajo sin servidor

Entorno de clúster tradicional

Muchos usuarios comienzan su traspaso a la nube con un entorno parecido a los entornos de HPC tradicionales. El entorno a menudo implica un nodo de inicio de sesión con un programador para iniciar trabajos.

Un enfoque común para el aprovisionamiento de clúster tradicional se basa en una plantilla de AWS CloudFormation para un clúster de informática combinado con la personalización para las tareas específicas de un usuario. AWS ParallelCluster es un ejemplo de una capacidad de aprovisionamiento de clústeres de extremo a extremo basada en AWS CloudFormation. Si bien la descripción compleja de la arquitectura está oculta dentro de la plantilla, las opciones típicas de configuración permiten que el usuario seleccione el tipo de instancia, el programador o las acciones de arranque, como la instalación de aplicaciones o la sincronización de datos. La plantilla se puede crear y ejecutar para brindar un entorno de HPC con el "aspecto" de los clústeres convencionales de HPC, aunque se agrega el beneficio de escalabilidad. El nodo de inicio de sesión mantiene el programador, el sistema de archivos compartidos y el entorno de ejecución. Mientras tanto, un mecanismo de ajuste de escala automático permite que más instancias giren a medida que los trabajos se envían a una cola de trabajos. A medida que las instancias quedan inactivas, se terminan de manera automática.

Un clúster puede implementarse en una configuración persistente o tratarse como un recurso efímero. Los clústeres persistentes se implementan con una instancia de inicio de sesión y una flota de cómputos que puede ser de tamaño fijo o estar vinculada a un grupo de Auto Scaling que aumenta y disminuye la flota de cómputos, según la cantidad de trabajos enviados. Los clústeres persistentes siempre tienen infraestructura funcionando. Alternativamente, los clústeres se pueden tratar como efímeros, donde cada carga de trabajo se ejecuta en su propio clúster. La automatización habilita los clústeres efímeros. Por ejemplo, un script de bash se combina con la CLI de AWS, o un script de Python con un SDK de AWS proporciona

automatización de casos de extremo a extremo. Para cada caso, se aprovisionan y lanzan recursos, los datos se colocan en los nodos, los trabajos se ejecutan en varios nodos y la salida de casos se recupera automáticamente o se envía a Amazon S3. Tras completar el trabajo, se termina la infraestructura. Estos clústeres tratan la infraestructura como código, optimizan los costes y permiten controlar completamente la versión de los cambios de infraestructura.

Las arquitecturas de clústeres tradicionales se pueden utilizar para cargas de trabajo levemente y estrechamente acopladas. Para mejorar el rendimiento, las cargas de trabajo estrechamente acopladas deben utilizarse como una flota de cómputos en un grupo de ubicación en clúster con tipos de instancia homogéneos.

Arquitectura de referencia

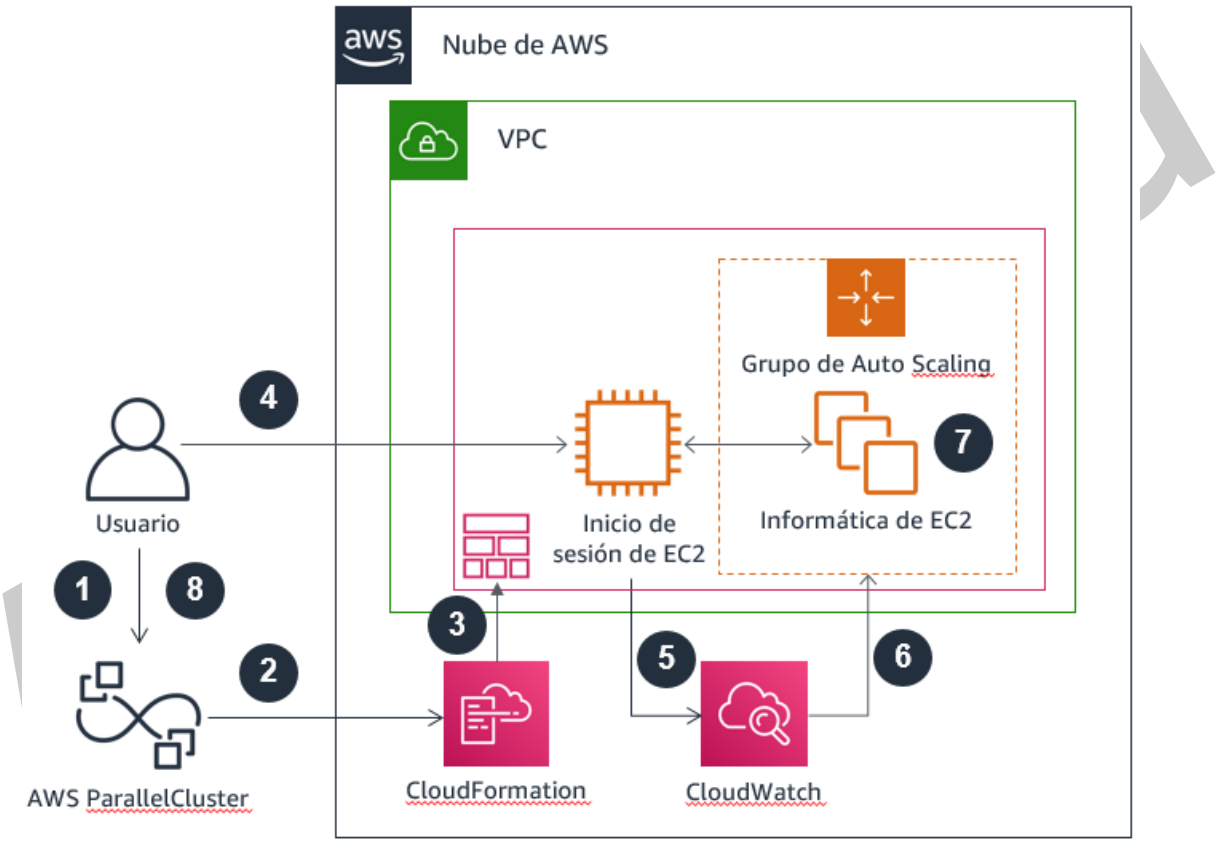

Figura 1: clúster tradicional implementado con AWS ParallelCluster

Pasos del flujo de trabajo:

1. El usuario inicia la creación de un clúster a través de la CLI y la especificación de AWS ParallelCluster en el archivo de configuración.
2. AWS CloudFormation crea la arquitectura del clúster tal y como se describe en el archivo de plantilla del clúster, en el que el usuario aportó unos pocos ajustes

personalizados (por ejemplo, la edición de un archivo de configuración o el uso de una interfaz web).

3. AWS CloudFormation implementa la infraestructura de instantáneas de EBS creada con aplicaciones/software personalizados de HPC a los que las instancias de clúster pueden acceder a través de una exportación NFS.
4. El usuario inicia sesión en la instancia de inicio de sesión y envía trabajos al programador (por ejemplo, SGE, Slurm).
5. La instancia de inicio de sesión emite métricas a CloudWatch según el tamaño de la cola del trabajo.
6. CloudWatch activa eventos de Auto Scaling para aumentar la cantidad de instancias de informática si el tamaño de la cola del trabajo supera un límite.
7. Los trabajos programados son procesados por la flota de cómputo.
8. [Opcional] El usuario inicia la eliminación del clúster y la terminación de todos los recursos.

Arquitectura basada en lotes

AWS Batch es un servicio totalmente administrado que permite ejecutar cargas de trabajo informáticas a gran escala en la nube sin aprovisionar recursos o administrar programadores.[3] AWS Batch permite a los desarrolladores, científicos e ingenieros ejecutar de manera fácil y eficiente cientos de miles de trabajos informáticos por lote en AWS. AWS Batch aprovisiona dinámicamente la cantidad y el tipo óptimos de recursos informáticos (por ejemplo, CPU o instancias con optimización de memoria) en función del volumen y de los requisitos de recursos específicos de los trabajos por lotes enviados. Planifica, programa y ejecuta sus cargas de trabajo de informática por lotes en toda la gama de características y servicios informáticos de AWS, como Amazon EC2[4] e instancia de spot.[5] Puede enfocarse en analizar los resultados y obtener nuevos conocimientos, sin tener que instalar y administrar el software de informática por lotes ni los clústeres de servidores necesarios para ejecutar sus trabajos.

Con AWS Batch, empaqueta su aplicación en un contenedor, especifica las dependencias de su trabajo y envía los trabajos en lote mediante la consola de administración de AWS, la CLI o un SDK. Puede especificar parámetros de ejecución y dependencias de trabajo e integrarse con una amplia gama de lenguajes y motores de flujo de trabajo de informática por lotes populares (por ejemplo, Pegasus WMS, Luigi y AWS Step Functions). AWS Batch ofrece colas de trabajo predeterminadas y definiciones de entorno de cómputo que permiten comenzar con rapidez.

Se puede utilizar una arquitectura basada en AWS Batch para cargas de trabajo levemente y estrechamente acopladas. Las cargas de trabajo estrechamente acopladas deben utilizar trabajos en paralelo con varios nodos en AWS Batch.

Arquitectura de referencia

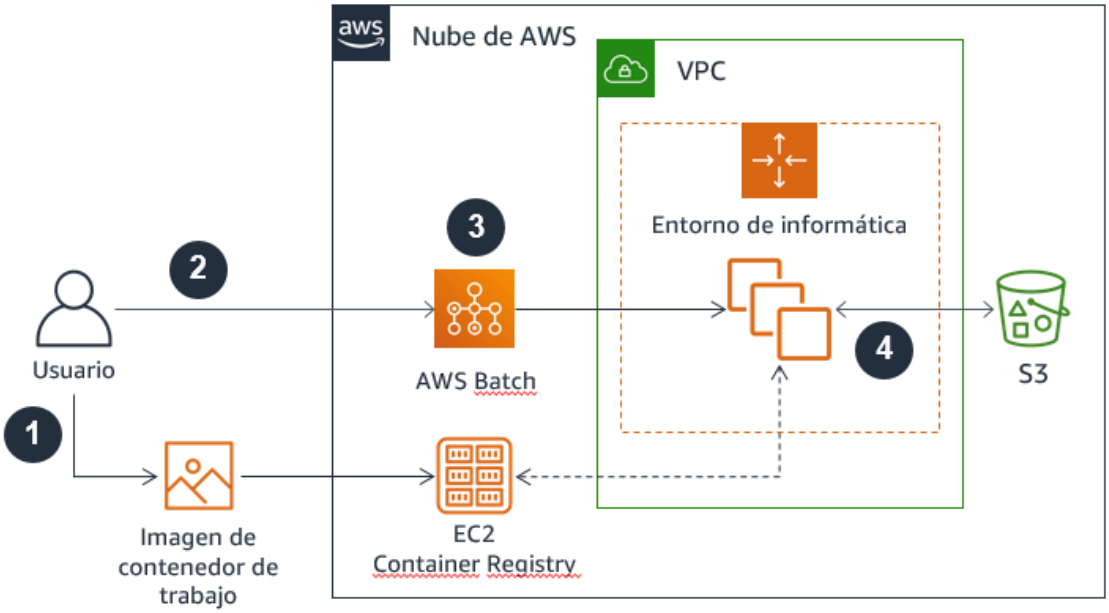

Figura 2: ejemplo de arquitectura de AWS Batch

Pasos del flujo de trabajo:

1. El usuario crea un contenedor de trabajo, carga el contenedor en Amazon EC2 Container Registry (Amazon ECR) u otro registro de contenedor (por ejemplo, DockerHub) y crea una definición de trabajo en AWS Batch.
2. El usuario envía los trabajos a una cola de trabajo en AWS Batch.
3. AWS Batch extrae la imagen del registro de contenedor y procesa los trabajos en la cola
4. En S3 bucket, se almacenan los datos de entrada y salida de cada trabajo.

Arquitectura basada en colas

Amazon SQS es un servicio de cola de mensajes totalmente administrado que facilita el desacople de los pasos de preprocesamiento de los pasos de cómputo y los pasos posprocesamiento.[6] La creación de aplicaciones a partir de componentes individuales que realizan funciones discretas mejora la escalabilidad y la confiabilidad. El desacople de componentes es una práctica recomendada para diseñar aplicaciones modernas. A menudo, Amazon SQS ocupa el lugar central de las soluciones levemente acopladas nativas en la nube.

Amazon SQS suele estar organizado con las soluciones de scripts de AWS CLI o AWS SDK para implementar aplicaciones desde el escritorio sin que los usuarios interactúen directamente con componentes de AWS. Una arquitectura basada en colas con SQS y EC2 exige una

infraestructura de cómputo autoadministrada, que difiere de una implementación administrada por servicio, como AWS Batch.

Una arquitectura basada en colas es mejor para las cargas levemente acopladas y puede volverse rápidamente compleja si se aplica a cargas de trabajo estrechamente acopladas.

Arquitectura de referencia

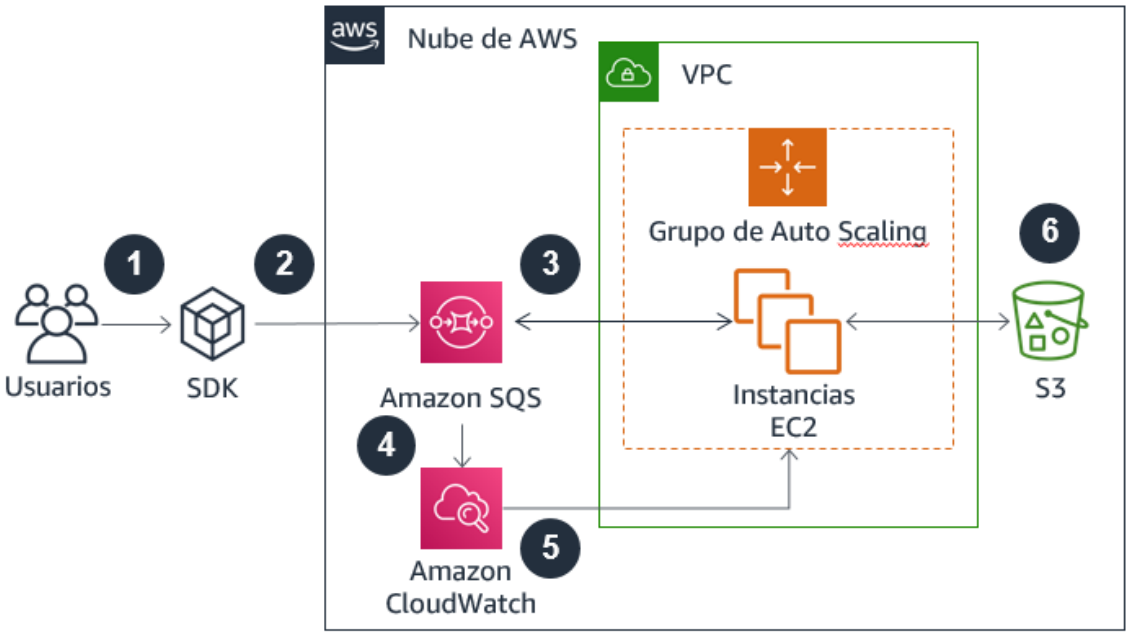

Figura 3: implementación de Amazon SQS para una carga de trabajo levemente acoplada

Pasos del flujo de trabajo:

1. Varios usuarios envían trabajos con la CLI o el SDK de AWS.
2. Los trabajos se ponen en cola como mensajes en Amazon SQS.
3. Las instancias EC2 sondean la cola e inician el procesamiento de los trabajos.
4. Amazon SQS emite métricas según la cantidad de mensajes (trabajos) en la cola.
5. Se configura una alarma de Amazon CloudWatch para avisar a Auto Scaling si la cola es más extensa que la longitud especificada. Auto Scaling aumenta la cantidad de instancias EC2.
6. Las instancias EC2 sondean los datos de origen y almacenan datos de resultado en un bucket de S3.

Implementación híbrida

Las implementaciones híbridas son consideradas en principio por las organizaciones que invierten en su infraestructura en las instalaciones y que también desean utilizar AWS. Este enfoque permite a las organizaciones aumentar los recursos en las instalaciones y crea una ruta alternativa a AWS en lugar de una migración completa inmediata.

Los escenarios híbridos varían desde coordinación mínima, como la separación de la carga de trabajo, hasta enfoques estrechamente integrados, como la colocación de trabajos impulsada por el programador. Por ejemplo, es posible que una organización separe todas sus cargas de trabajo y las ejecute de un determinado tipo en la infraestructura de AWS. Como alternativa, es posible que las organizaciones con una gran inversión en sus procesos e infraestructura en las instalaciones deseen obtener una experiencia más fluida para sus usuarios finales cuando administran los recursos de AWS con su software de programación de trabajos y, posiblemente, un portal de envío de trabajos. Varios programadores de trabajo (comerciales y de código abierto) ofrecen la capacidad para aprovisionar y desaprovisionar dinámicamente los recursos de AWS según sea necesario. La administración subyacente de recursos depende de las integraciones de AWS nativas (por ejemplo, la CLI o API de AWS) y pueden permitir un entorno altamente personalizado, según el programador. Si bien los programadores de trabajo ayudan a administrar los recursos de AWS, el programador es solo un aspecto para lograr una implementación exitosa.

Los factores fundamentales para operar de manera exitosa un escenario híbrido son la localidad y el movimiento de los datos. Algunas cargas de trabajo de HPC no exigen ni generan bases de datos importantes. Por lo tanto, la administración de datos es menos preocupante. Sin embargo, los trabajos que exigen grandes cantidades de datos de entrada, o que generan importantes datos de salida, pueden convertirse en un cuello de botella. Las técnicas para abordar la administración de datos varían según la organización. Por ejemplo, una organización puede hacer que sus usuarios finales administren la transferencia de datos en sus scripts de envío de trabajo, otras pueden ejecutar solo determinados trabajos en la ubicación en la que se encuentre un conjunto de datos, una organización de terceros puede elegir duplicar los datos en ambos lugares, y otra organización puede elegir utilizar una combinación de varias opciones.

Según el enfoque de administración de datos, AWS ofrece varios servicios para asistir en una implementación de soluciones híbridas. Por ejemplo, AWS Direct Connect establece una conexión de red dedicada entre el entorno en las instalaciones y AWS, y AWS DataSync migra de manera automática los datos desde el almacenamiento en las instalaciones a Amazon S3 o Amazon Elastic File System. Las empresas externas disponen de opciones de software adicionales en AWS Marketplace y la red de socios de AWS (APN).

Las arquitecturas de implementación híbrida se pueden utilizar para cargas de trabajo levemente y estrechamente acopladas. Sin embargo, para mejorar el rendimiento, una única carga de trabajo estrechamente acoplada debe encontrarse en las instalaciones o en AWS.

Arquitectura de referencia

Figura 3: ejemplo de implementación híbrida basada en el programador

Pasos del flujo de trabajo:

1. El usuario envía el trabajo a un programador (por ejemplo, Slurm) en un nodo de inicio de sesión en las instalaciones.
2. El programador ejecuta el trabajo en el cómputo en las instalaciones o en la infraestructura de AWS según la configuración.
3. Los trabajos acceden al almacenamiento compartido según su ubicación de ejecución.

Sin servidor

El traspaso a la nube levemente acoplado a menudo provoca un entorno que no tiene ningún servidor, esto significa que puede concentrarse en sus aplicaciones y dejar la responsabilidad del aprovisionamiento del servidor a los servicios administrados. AWS Lambda puede ejecutar códigos sin aprovisionar ni administrar servidores. Solo paga el tiempo de procesamiento que consume, sin ningún cargo mientras su código no se ejecuta. Carga su código y Lambda se ocupará de todo lo necesario para ejecutarlo y escalarlo. Además, Lambda tiene la capacidad de activar de manera automática eventos de otros servicios de AWS.

La escalabilidad es una segunda ventaja del enfoque sin servidor de Lambda. Si bien cada trabajador puede ser de tamaño modesto, por ejemplo, un núcleo de cómputo con algo de memoria, la arquitectura puede generar miles de trabajadores de Lambda simultáneos, y así alcanzar una gran capacidad de rendimiento de cómputos y obtener la etiqueta de HPC. Por ejemplo, se puede analizar una gran cantidad de archivos mediante invocaciones del mismo algoritmo, se puede analizar una gran cantidad de genomas en paralelo o se puede modelar una gran cantidad de sitios de genes dentro de un genoma. La mayor escala y velocidad de escalado alcanzables. Si bien las arquitecturas basadas en servidores necesitan un tiempo del orden de los minutos para aumentar la capacidad de respuesta a una solicitud (incluso cuando se utilizan máquinas virtuales como las instancias EC2), las funciones de Lambda sin servidor se escalan en segundos. AWS Lambda habilita la infraestructura de HPC que responde de inmediato a cualquier solicitud imprevista de resultados intensivos de cómputo y puede cumplir con un número variable de solicitudes sin que se prevea un derroche de recursos.

Además de los cómputos, existen otras arquitecturas sin servidor que ayudan a los flujos de trabajo de HPC. AWS Step Functions permite coordinar varios pasos en una canalización mediante la unión de diferentes servicios de AWS. Por ejemplo, se puede crear una canalización de genómica con AWS Step Functions para la coordinación, Amazon S3 para el almacenamiento, AWS Lambda para pequeñas tareas y AWS Batch para el procesamiento de datos.

Las arquitecturas sin servidor son las mejores para las cargas de trabajo levemente acopladas o como coordinación de flujos de trabajo si se combinan con otra arquitectura de HPC.

Arquitectura de referencia

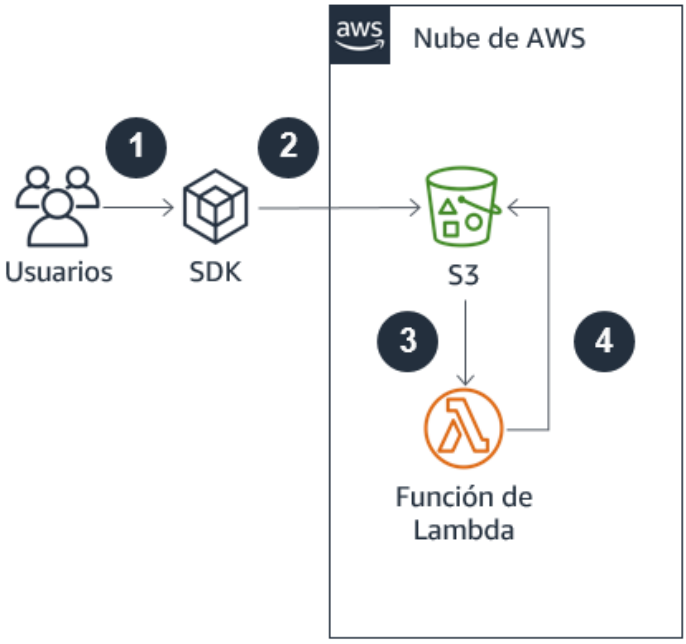

Figura 4: ejemplo de carga de trabajo levemente acoplada implementada por Lambda

Pasos del flujo de trabajo:

1. El usuario carga un archivo a un bucket de S3 a través de la CLI o el SDK de AWS.
2. El archivo de entrada se guarda con un prefijo de entrada (por ejemplo, input/).
3. Un evento de S3 activa de manera automática una función de Lambda para procesar los datos entrantes.
4. El archivo de salida se guarda nuevamente en el bucket de S3 con un prefijo de salida (por ejemplo, output/.)

Los cinco pilares del marco de buena arquitectura

Esta sección describe la HPC en el contexto de los cinco pilares del marco de buena arquitectura. Cada pilar analiza los principios de diseño, las definiciones, las prácticas recomendadas, las preguntas de evaluación, las consideraciones, los servicios de AWS clave y los enlaces útiles.

Pilar de excelencia operativa

El pilar de **excelencia operativa** incluye la capacidad de ejecutar y monitorear sistemas para ofrecer valor empresarial, y mejorar continuamente los procesos y procedimientos de soporte.

Principios de diseño

En la nube, una serie de principios impulsan la excelencia operativa. En particular, se enfatizan los siguientes para las cargas de trabajo de HPC. También consulte los principios de diseño en el documento técnico del Marco de Buena Arquitectura de AWS.

- **Automatizar las operaciones de los clústeres:** en la nube, puede definir toda su carga de trabajo como código y actualizarla con código. De esta manera, puede automatizar los procesos y procedimientos repetitivos. Obtiene beneficios de poder reproducir de manera sistemática la infraestructura e implementar los procedimientos operativos. Esto incluye la automatización del proceso de envío de trabajos y las respuestas a eventos, como el inicio, la finalización o el fallo del trabajo. En la HPC, es normal que los usuarios esperen repetir varios pasos para cada trabajo incluidos, por ejemplo, la carga de archivos de caso, el envío de un trabajo a un programador y la transferencia de los archivos de resultados. Automatice estos pasos repetitivos con scripts o mediante código controlado por eventos para maximizar la usabilidad y minimizar los costes y los errores.
- **Utilizar arquitecturas nativas en la nube cuando corresponda:** las arquitecturas de HPC suelen adoptar una de dos formas. La primera es una configuración tradicional de clúster con una instancia de inicio de sesión, nodos de computación y un programador de trabajos. La segunda es una arquitectura nativa en la nube con implementaciones automatizadas y servicios administrados. Se puede ejecutar una única carga de trabajo para cada clúster (efímero) o utilizar capacidades sin servidor. Las arquitecturas nativas en la nube pueden optimizar las operaciones con la democratización de tecnologías avanzadas. Sin embargo, el mejor enfoque tecnológico se alinea con el entorno deseado para los usuarios de HPC.

Definición

Existen tres áreas de prácticas recomendadas para la excelencia operativa en la nube:

- Preparación
- Operación
- Evolución

Para obtener más información sobre la **preparación**, **operación** y **evolución** de las áreas, consulte el documento técnico del Marco de Buena Arquitectura de AWS. En este documento técnico no se describe la evolución.

Prácticas recomendadas

Preparación
Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

A medida que se prepara para implementar su carga de trabajo, considere la posibilidad de utilizar paquetes especializados de software (comerciales o de código abierto) para obtener visibilidad en la información del sistema y aprovechar esta información para definir los patrones de arquitectura para sus cargas de trabajo. Utilice herramientas de automatización, como AWS ParallelCluster o AWS CloudFormation, para definir estas arquitecturas de manera tal que se puedan configurar con las variables.

La nube ofrece varias opciones de programación. Una opción es utilizar AWS Batch, que es un servicio de procesamiento en lote completamente administrado con compatibilidad para tareas tanto de un solo nodo como de varios nodos. Otra opción es no utilizar un programador. Por ejemplo, puede crear un clúster efímero para ejecutar un único trabajo directamente.

> HPCOPS 1: ¿Cómo estandariza las arquitecturas en los clústeres?
>
> HPCOPS 2: ¿Cómo programa trabajos: programadores tradicionales, AWS Batch o ningún programador con clústeres efímeros?

Operación
Las operaciones deben estandarizarse y administrarse de manera rutinaria. Concéntrese en la automatización, los pequeños cambios frecuentes, las pruebas regulares de garantía de calidad y los mecanismos definidos para rastrear, auditar, revertir y revisar los cambios. Los cambios no deben ser grandes ni poco habituales, no deben exigir un tiempo de inactividad

programado ni deben requerir una ejecución manual. Se debe recopilar y revisar una amplia gama de registros y métricas basados en indicadores operativos clave para una carga de trabajo, a fin de garantizar la continuidad de las operaciones.

AWS ofrece la oportunidad de utilizar herramientas adicionales para manejar operaciones de HPC. Estas herramientas pueden variar desde el monitoreo de la asistencia hasta la automatización de las implementaciones. Por ejemplo, puede hacer que Auto Scaling reinicie las instancias fallidas, usar CloudWatch para monitorear las métricas de carga de su clúster, configurar notificaciones para cuando los trabajos finalicen o usar un servicio administrado (como AWS Batch) para implementar reglas de reintento para los trabajos fallidos. Las herramientas nativas en la nube pueden mejorar en gran medida la implementación de sus aplicaciones y la administración de sus cambios.

Los procesos de administración de versiones, ya sean manuales o automatizados, deben basarse en pequeños cambios incrementales y en el seguimiento de las versiones. Debe poder revertir las versiones que causan problemas sin provocar un impacto operativo. Utilice herramientas de integración e implementación continuas, como AWS CodePipeline y AWS CodeDeploy, para automatizar la implementación de cambios. Realice un seguimiento de los cambios en el código de origen con herramientas de control de versiones, como AWS CodeCommit, y de las configuraciones de la infraestructura con herramientas de automatización, como las plantillas de AWS CloudFormation.

HPCOPS 3: ¿Cómo evoluciona su carga de trabajo mientras minimiza el impacto del cambio?

HPCOPS 4: ¿Cómo monitorea su carga de trabajo para asegurarse de que funciona según lo previsto?

El uso de la nube para HPC introduce nuevas consideraciones operativas. Si bien los clústeres en las instalaciones se fijan en tamaño, los clústeres en la nube pueden escalar para satisfacer la demanda. Además, las arquitecturas nativas en la nube para HPC no funcionan igual que las arquitecturas en las instalaciones. Por ejemplo, utilizan mecanismos diferentes para enviar trabajos y aprovisionar los recursos de la instancia bajo demanda a medida que llegan los trabajos. Puede adoptar procedimientos operativos que se adapten a la elasticidad de la nube y a la naturaleza dinámica de las arquitecturas nativas en la nube.

Evolución

No hay prácticas recomendadas únicas de HPC para el área de prácticas de **evolución**. Para obtener más información, consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Pilar de seguridad

El pilar de **seguridad** incluye la capacidad de proteger la información, los sistemas y los recursos al mismo tiempo que ofrece valor de negocio mediante evaluaciones de riesgos y estrategias de mitigación.

Principios de diseño

En la nube, existe una serie de principios que lo ayudan a fortalecer la seguridad de su sistema. Se recomiendan los principios de diseño del documento técnico de Marco de Buena Arquitectura de AWS porque no varían para las cargas de trabajo de HPC.

Definición

Existen cinco áreas de prácticas recomendadas para la seguridad en la nube:

- Identity and Access Management (IAM)
- Controles de detección
- Protección de la infraestructura
- Protección de los datos
- Respuesta ante incidentes

Antes de diseñar cualquier sistema, debe establecer prácticas de seguridad. Debe poder controlar los permisos, identificar los incidentes de seguridad, proteger sus sistemas y servicios, y mantener la confidencialidad e integridad de los datos a través de la protección de datos. Debe tener un proceso ben definido y practicado para responder a los incidentes de seguridad. Estas herramientas y técnicas son importantes porque respaldan los objetivos, como la prevención de la pérdida de datos y el cumplimiento con las obligaciones reglamentarias.

El modelo de responsabilidad compartida de AWS permite que las organizaciones adopten la nube para lograr sus metas de seguridad y cumplimiento. Debido a que AWS asegura físicamente la infraestructura que es compatible con nuestros servicios en la nube, puede enfocarse en utilizar servicios para lograr sus metas. La nube de AWS ofrece acceso a los datos de seguridad y un enfoque automatizado para responder a los eventos de seguridad.

Todas las áreas de las prácticas recomendadas de seguridad son vitales y están bien documentadas en el documento técnico del Marco de Buena Arquitectura de AWS. En el documento técnico del Marco de Buena Arquitectura de AWS se describen las áreas de **controles de detección**, **protección de la infraestructura** y **respuesta ante incidentes**. No se describen en este informe ni requieren modificación para las cargas de trabajo de HPC.

Prácticas recomendadas

Identity and Access Management (IAM)
Identity and Access Management son partes clave de un programa de seguridad de la información. Aseguran que solo los usuarios con autorización y autenticación accedan a los recursos. Por ejemplo, define las entidades principales (usuarios, grupos, servicios y funciones que intervienen en su cuenta), construye políticas que hacen referencia a estas entidades e implementa una fuerte administración de credenciales. Estos elementos de administración de privilegios constituyen los conceptos centrales de la autenticación y la autorización.

Ejecute cargas de trabajo de HPC de manera autónoma y efímera para limitar la exposición de la información confidencial. Las implementaciones autónomas exigen poco acceso humano a las instancias, lo que minimiza la exposición de los recursos. Los datos de HPC se producen en un tiempo limitado, esto minimiza la posibilidad de un posible acceso no autorizado a los datos.

> HPCSEC 1: ¿Cómo utiliza los servicios administrados, los métodos autónomos y los clústeres efímeros para minimizar el acceso humano a la infraestructura de la carga de trabajo?

Las arquitecturas de HPC pueden utilizar una variedad de servicios de cómputos administrados (por ejemplo, AWS Batch, AWS Lambda, etc.) y no administrados (por ejemplo, Amazon EC2). Cuando las arquitecturas necesitan un acceso directo a los entornos de informática, como la conexión a una instancia EC2, los usuarios suelen conectarse a través de un Secure Shell (SSH) y se autentican con una clave SSH. Este modelo de acceso es habitual en una escenario de clúster tradicional. Todas las credenciales, como las claves SSH, deben protegerse de manera adecuada y cambiarse con regularidad.

Como alternativa, AWS Systems Manager cuenta con un servicio completamente administrado (Session Manager) que ofrece un shell interactivo basado en navegador y una experiencia de CLI. Ofrece una administración de instancias segura y auditable sin abrir los puertos de entrada, mantener los hosts bastión y administrar las claves SSH. Se puede acceder a Session Manager mediante algún cliente SSH que sea compatible con ProxyCommand.

> HPCSEC 2: ¿Qué métodos utiliza para proteger y administrar sus credenciales?

Controles de detección
No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas de controles de detección. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Protección de la infraestructura
No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas de la infraestructura. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Protección de los datos
Antes de diseñar cualquier sistema, debe establecer prácticas de seguridad fundamentales. Por ejemplo, la clasificación de datos ofrece una manera de categorizar los datos de la organización según los niveles de sensibilidad, mientras que el cifrado protege los datos haciéndolos ininteligibles para el acceso no autorizado. Estas herramientas y técnicas son importantes porque respaldan los objetivos ,como la prevención de la pérdida de datos o el cumplimiento con las obligaciones reglamentarias.

> HPCSEC 3: ¿Cómo aborda su arquitectura los requisitos de datos para la disponibilidad y la durabilidad del almacenamiento a lo largo del ciclo de vida de sus resultados?

Además del nivel de sensibilidad y las obligaciones reglamentarias, los datos de HPC también pueden categorizarse según el momento y la manera en que se volverán a usar. A menudo se conservan los resultados finales, mientras que los resultados intermedios, que pueden volver a crearse si es necesario, pueden no necesitar conservarse. La evaluación y categorización cuidadosa de los datos permite la migración programática de datos importantes a soluciones de almacenamiento más resistentes, como Amazon S3 y Amazon EFS.

Una comprensión de la duración de los datos en combinación con el manejo programático de estos ofrece la mínima exposición y la máxima protección de la infraestructura de buena arquitectura.

Respuesta ante incidentes
No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas para respuesta ante incidentes. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Pilar de confiabilidad

El pilar de **confiabilidad** incluye la capacidad de un sistema para recuperarse de las interrupciones de la infraestructura o del servicio, adquirir dinámicamente recursos informáticos para satisfacer la demanda y mitigar interrupciones como las configuraciones incorrectas o los problemas transitorios de red.

Principios de diseño

En la nube, una serie de principios lo ayudan a aumentar la confiabilidad. En particular, se enfatizan los siguientes para las cargas de trabajo de HPC. Para obtener más información,

consulte los principios de diseño en el documento técnico del Marco de Buena Arquitectura de AWS.

- **Escalar de manera horizontal para aumentar la disponibilidad agregada del sistema:** es importante considerar las opciones de escala horizontal que pueden reducir el efecto de un solo error en todo el sistema. Por ejemplo, en lugar de tener un gran clúster de HPC compartido que ejecute varios trabajos, considere la posibilidad de crear varios clústeres en la infraestructura de Amazon para aislar aún más su riesgo de que se produzcan posibles errores. Debido a que la infraestructura se puede tratar como código, puede escalar de manera horizontal los recursos dentro de un único clúster, así como la cantidad de clústeres que ejecutan determinados casos.
- **Dejar de adivinar la capacidad:** se puede aprovisionar un conjunto de clústeres de HPC para satisfacer las necesidades actuales y escalar manual o automáticamente para cumplir con los aumentos o las disminuciones de la demanda. Por ejemplo, termine los nodos de computación inactivos cuando no se usen y ejecútelos. Por ejemplo, termine los nodos de computación inactivos cuando no se usen y ejecute clústeres simultáneos para procesar varios cálculos en lugar de esperar en una cola.
- **Administrar cambios en la automatización:** la automatización de cambios en la infraestructura permite colocar una infraestructura de clúster bajo control de versiones y hacer duplicados exactos de un clúster ya creado. Se deben administrar los cambios de la automatización.

Definición

Existen tres áreas de prácticas recomendadas para la confiabilidad en la nube:

- Fundamentos
- Administración de cambios
- Administración de errores

El área de administración de cambios se describe en el documento técnico del Marco de Buena Arquitectura de AWS.

Prácticas recomendadas

Fundamentos

HPCREL 1: ¿Cómo administra los límites de servicio de AWS de sus cuentas?

AWS establece límites de servicio (un límite superior en el número de cada recurso que su equipo puede solicitar) para protegerlo contra un aprovisionamiento excesivo de recursos accidental.

Las aplicaciones de HPC suelen necesitar una gran cantidad de instancias de informática en simultáneo. La capacidad y las ventajas de la escala horizontal son sumamente convenientes para las cargas de trabajo de HPC. Sin embargo, la escala horizontal puede necesitar un aumento de los límites de servicio de AWS antes de que se implemente una gran carga de trabajo en un gran grupo o en muchos grupos más pequeños a la vez.

A menudo, se deben aumentar los límites de servicio con respecto a los valores predeterminados a fin de manejar los requisitos de una gran implementación. Para solicitar un aumento, comuníquese con AWS Support.

Administración de cambios

No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas de la administración de cambios. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Administración de errores

Todos los sistemas complejos pueden esperar que ocurran errores ocasionales, por lo que es fundamental conocerlos, responder ante ellos y evitar que vuelvan a ocurrir. Los escenarios de error pueden incluir el fallo de un clúster en el arranque o el fallo de una carga de trabajo específica.

> HPCREL 2: ¿Cómo utiliza su aplicación los puntos de comprobación para recuperarse de los errores?

Se puede mejorar la tolerancia a los errores de varias maneras. Para los casos de larga duración, incorporar puntos de comprobación regulares en su código permite continuar desde un estado parcial en caso de error. Los puntos de comprobación son una característica habitual de la administración de errores de la aplicación que ya está incorporada en muchas aplicaciones de HPC. El enfoque más habitual es que las aplicaciones escriban de manera periódica los resultados intermedios. Los resultados intermedios ofrecen posibles conocimientos de los errores de la aplicación y la capacidad de reiniciar el caso según sea necesario, mientras que solo se pierde una parte del trabajo.

Los puntos de comprobación son útiles en las instancias de spot cuando utiliza instancias muy rentables, pero posiblemente interrumpibles. Además, algunas aplicaciones pueden beneficiarse del cambio de comportamiento predeterminado de la interrupción de spot (por ejemplo, detener o hibernar la instancia en lugar de terminarla). Es importante tener en cuenta la durabilidad de la opción de almacenamiento cuando confía en los puntos de comprobación para administrar los errores.

HPCREL 3: ¿Cómo planificó la tolerancia a los errores en su arquitectura?

Se puede mejorar tolerancia a los errores cuando se implementa en varias zonas de disponibilidad. Los requisitos de baja latencia de las aplicaciones de HPC estrechamente acopladas necesitan que cada caso se encuentre en un grupo único de ubicación en clúster y en una zona de disponibilidad. Por otra parte, las aplicaciones levemente acopladas no tienen esos requisitos de baja latencia y pueden mejorar la administración de errores con la capacidad de implementarse en varias zonas de disponibilidad.

Tenga en cuenta la compensación entre los pilares de confiabilidad y coste cuando tome esta decisión de diseño. La duplicación del almacenamiento y la infraestructura de informática (por ejemplo, un nodo principal y el almacenamiento asociado) implica otro costo, por lo que es posible que haya cargos de transferencia de datos cuando migre los datos a una zona de disponibilidad o a otra región de AWS. Para casos de uso no urgentes, puede ser preferible migrar a otra zona de disponibilidad como parte de un evento de recuperación ante desastres (DR).

Pilar de eficiencia del rendimiento

El pilar de **eficiencia del rendimiento** se centra en el uso eficiente de los recursos de computación para satisfacer los requisitos y el mantenimiento de dicha eficiencia a medida que la demanda cambia y las tecnologías evolucionan.

Principios de diseño

Cuando diseña para HPC en la nube, hay una serie de principios que lo ayudan a lograr la eficiencia del rendimiento:

- **Diseñar el clúster para la aplicación:** los clústeres tradicionales son estáticos y necesitan que la aplicación se diseñe para el clúster. AWS ofrece la capacidad de diseñar el clúster para la aplicación. Ya no se necesita un único modelo con clústeres individuales para cada aplicación. Cuando ejecuta una variedad de aplicaciones en AWS, se pueden utilizar distintas arquitecturas para satisfacer las demandas de la aplicación. De esta manera, se puede mejorar el rendimiento mientras se minimizan los costes.
- **Evaluar el rendimiento con un caso de uso significativo:** el mejor método para medir el rendimiento de una aplicación de HPC en una arquitectura particular es la ejecución de una demostración significativa de la aplicación en sí. Un caso de demostración inadvertidamente pequeño o grande (uno sin el cómputo, la memoria, la transferencia de datos o el tráfico de red esperados) no ofrecerá una prueba significativa del rendimiento de la aplicación en AWS. Si bien los puntos de referencia específicos del sistema ofrecen

una comprensión del rendimiento de la infraestructura subyacente de cómputo, no reflejan la manera en que se comportará una aplicación en conjunto. El modelo de pago por uso de AWS hace que una prueba de concepto sea rápida y rentable.

- **Utilizar arquitecturas nativas en la nube cuando corresponda:** en la nube, las arquitecturas administradas, sin servidores y nativas en la nube eliminan la necesidad de ejecutar y mantener servidores para llevar a cabo actividades de cómputo tradicionales. Los componentes nativos en la nube para el cómputo específico de HPC, almacenamiento, organización de trabajos y de datos, así como de metadatos. La variedad de servicios de AWS permite que se desacople y optimice cada paso en el proceso de carga de trabajo para obtener una mayor capacidad de rendimiento.
- **Experimentar con frecuencia:** los recursos virtuales y automatizables permiten realizar pruebas comparativas con rapidez mediante diferentes tipos de instancias, almacenamiento y configuraciones.

Definición

Existen cuatro áreas de prácticas recomendadas para la eficiencia del rendimiento en la nube:

- Selección
- Revisión
- Monitoreo
- Compensaciones

En el documento técnico del Marco de Buena Arquitectura de AWS se describen las áreas de **revisión**, **monitoreo** y **compensaciones**.

Prácticas recomendadas

Selección

La solución óptima para un sistema particular varía según el tipo de carga de trabajo que tiene. Los sistemas de buena arquitectura utilizan varias soluciones y habilitan diferentes características para mejorar el rendimiento. Una arquitectura de HPC puede depender de uno o más elementos arquitectónicos diferentes, por ejemplo, en cola, en lote, en informática en clúster, en contenedores, sin servidor y basada en eventos.

Informática

HPCPERF 1: ¿Cómo selecciona su solución de informática?

La solución óptima de informática para una arquitectura particular de HPC depende del método de implementación de la carga de trabajo, el grado de automatización, los patrones de uso y la configuración. Es posible que se elijan diferentes soluciones de informática para cada paso de un proceso. Seleccionar las soluciones de informática incorrectas para una arquitectura puede disminuir la eficiencia del rendimiento.

Las instancias son servidores virtualizados, y se presentan en diferentes familias y tamaños para ofrecer una amplia variedad de capacidades. Algunas familias de instancias abordan cargas de trabajo específicas, por ejemplo, cargas de trabajo intensivas en informática, memoria o GPU. Otras instancias son de uso general.

Ambas familias de instancias de cargas de trabajo específicas y de uso general son útiles para las aplicaciones de HPC. Las instancias de particular interés para HPC incluyen la familia optimizada de informática y los tipos de instancias aceleradas, como GPU y FPGA.

Algunas familias de instancias ofrecen variantes dentro del grupo para obtener más capacidades. Por ejemplo, una familia de instancias puede tener una variante con almacenamiento local, mayores capacidades de redes o un procesador diferente. Estas variantes pueden visualizarse en la matriz de tipo de instancia[7] y pueden mejorar el rendimiento de algunas cargas de trabajo de HPC.

Dentro de cada familia de instancias, uno o más tamaños de instancias permiten la escala vertical de los recursos. Algunas aplicaciones exigen un tipo de instancia más grande (por ejemplo, 24xlarge), mientras que otras se ejecutan en tipos más pequeños (por ejemplo, large) según la cantidad o los procesos que son compatibles con la aplicación. El rendimiento óptimo se obtiene con el tipo de instancia más grande cuando se trabaja con una carga de trabajo estrechamente acoplada.

La familia de instancias de la serie T ha sido diseñada para aplicaciones con uso moderado de la CPU que pueden beneficiarse de una transmisión más allá del nivel básico de rendimiento de la CPU. La mayoría de las aplicaciones de HPC son intensivas en informática y sufren una disminución del rendimiento con la familia de instancias de la serie T.

Las aplicaciones varían en cuanto a sus requisitos (por ejemplo, núcleos deseados, velocidad del procesador, requisitos de la memoria, necesidades de almacenamiento y especificaciones de redes). Cuando seleccione un tipo y una familia de instancias, comience con las necesidades específicas de la aplicación. Los tipos de instancias pueden mezclarse y adaptarse a aplicaciones que necesiten instancias específicas para componentes de aplicación específicos.

Los **contenedores** son un método de virtualización del sistema operativo que resulta atractivo para muchas cargas de trabajo de HPC, en particular si las aplicaciones ya se han incluido en

contenedores. Los servicios de AWS como AWS Batch, Amazon Elastic Container Service (ECS) y Amazon Elastic Container Service for Kubernetes (EKS) ayudan a implementar las aplicaciones incluidas en contenedores.

Las **funciones** abstraen el entorno de ejecución. AWS Lambda permite ejecutar el código sin implementar, ejecutar ni mantener una instancia. Muchos servicios de AWS emiten eventos en función de la actividad dentro del servicio y, a menudo, se puede activar una función de Lambda de los eventos de servicio. Por ejemplo, se puede ejecutar una función de Lambda después de cargar un objeto en Amazon S3. Muchos usuarios de HPC utilizan Lambda para ejecutar de manera automática el código como parte de su flujo de trabajo.

Existen varias decisiones que tomar cuando lanza su instancia de informática selecta:

- **Sistema operativo:** para lograr el mejor rendimiento y asegurar el acceso a las bibliotecas más actualizadas, es fundamental un sistema operativo actual.
- **Tipo de virtualización:** las instancias EC2 de nueva generación se ejecutan en el sistema Nitro de AWS. El sistema Nitro ofrece todos los recursos de informática y de memoria del hardware anfitrión, lo que se traduce en un mejor rendimiento general. Las tarjetas Nitro dedicadas habilitan las redes de alta velocidad, el EBS de alta velocidad y la aceleración de E/S. Las instancias no retienen recursos para el software de administración.

El hipervisor Nitro es un hipervisor liviano que administra la memoria y la asignación de la CPU, y ofrece un rendimiento que no se distingue del nativo. El sistema Nitro también hace que las instancias nativas estén disponibles para ejecutarse sin el hipervisor Nitro. El lanzamiento de instancias nativas arranca el servidor subyacente, que incluye la verificación de todos los componentes de hardware y firmware. Esto significa que se puede demorar más antes de que la instancia nativa esté disponible para iniciar su carga de trabajo, en comparación con la instancia virtualizada. Se debe considerar el tiempo adicional de inicialización cuando opera en un entorno dinámico de HPC en el que los recursos se lanzan y terminan según la demanda.

HPCPERF 2: ¿Cómo optimiza el entorno de informática para su aplicación?

Características de hardware subyacentes: además de elegir una AMI, puede optimizar aún más su entorno si aprovecha las características del hardware de los procesadores de Intel subyacentes. Existen cuatro métodos principales a tener en cuenta cuando se optimiza el hardware subyacente:

1. Características avanzadas del procesador
2. Tecnología Hyper-Threading de Intel

3. Afinidad del procesador
4. Control de estado del procesador

Las aplicaciones de HPC pueden beneficiarse de estas características avanzadas del procesador (por ejemplo, las extensiones vectoriales avanzadas) y pueden aumentar sus velocidades de cálculo si compilan el software para la arquitectura de Intel.[8] Las opciones del compilador para las instrucciones específicas de la arquitectura varían según el compilador (consulte la guía de uso para su compilador).

AWS habilita la tecnología Hyper-Threading de Intel, comúnmente conocida como "hyperthreading", de forma predeterminada. El hyperthreading mejora el rendimiento de algunas aplicaciones con un proceso por hyperthread (dos procesos por núcleo). La mayoría de las aplicaciones de HPC se benefician de la desactivación de hyperthreading y, por lo tanto, tiende a ser el entorno preferido para las aplicaciones de HPC. El hyperthreading se deshabilita fácilmente en Amazon EC2. A menos que se haya probado una aplicación con hyperthreading habilitado, se recomienda que se deshabilite y que los procesos se lancen y fijen individualmente a los núcleos cuando se ejecuten aplicaciones de HPC. La afinidad de la CPU o el procesador permite que el proceso se fije con facilidad.

La afinidad del procesador se puede controlar de distintas maneras. Por ejemplo, se puede configurar a nivel del sistema operativo (disponible en Windows y Linux), establecer como un indicador de compilación dentro de la biblioteca de threading o especificar como un indicador MPI durante la ejecución. El método seleccionado para controlar la afinidad de los procesadores depende de su carga de trabajo y de la aplicación.

AWS permite ajustar el control de estado del procesador en determinados tipos de instancias.[9] Puede considerar la alteración de los ajustes de los estados C (estados inactivos) y los estados P (estados operativos) para optimizar su rendimiento. Los ajustes predeterminados del estado C y P ofrecen máximo rendimiento, que es fundamental para la mayoría de las cargas de trabajo. Sin embargo, si su aplicación se beneficiara de la reducción de la latencia a costa de frecuencias más altas de uno o dos núcleos, o del rendimiento constante a frecuencias más bajas en contraposición a las frecuencias picos de Turbo Boost, experimente con los ajustes de estados C o P disponibles en las instancias selectas.

Hay muchas opciones de informática para optimizar un entorno de informática. La implementación en la nube permite la experimentación en todos los niveles, desde el sistema operativo hasta el tipo de instancia y las implementaciones nativas. Debido a que los clústeres estáticos se ajustan antes de la implementación, el tiempo dedicado a experimentar con clústeres basados en la nube es vital para lograr el rendimiento deseado.

Almacenamiento

HPCPERF 3: ¿Cómo selecciona su solución de almacenamiento?

La solución óptima de almacenamiento para una arquitectura particular de HPC depende en gran medida de cada aplicación específica para esa arquitectura. También se consideran factores el método de implementación de cargas de trabajo, el grado de automatización y los patrones del ciclo de vida de los datos deseados. AWS ofrece una amplia gama de opciones de almacenamiento. Al igual que la informática, el mejor rendimiento se obtiene cuando se abordan las necesidades específicas de almacenamiento de una aplicación. AWS no necesita que sobreaprovisione su almacenamiento para un enfoque "único", y no siempre se necesitan sistemas de archivos compartidos grandes o de alta velocidad. Optimizar la selección de cómputo es importante para mejorar el rendimiento de HPC, pero muchas aplicaciones de HPC no se beneficiarán de la solución de almacenamiento más rápida posible.

Las implementaciones de HPC suelen necesitar un sistema de archivos compartidos y de alto rendimiento al que se acceda mediante nodos de informática en clústeres. Existen varios patrones de arquitectura que puede utilizar para implementar estas soluciones de almacenamiento de AWS Managed Services, las ofertas de AWS Marketplace, las soluciones de los socios de APN y las configuraciones de código abierto implementadas en las instancias EC2. En particular, Amazon FSx for Lustre es un servicio administrado que brinda una solución rentable y eficaz para las arquitecturas de HPC que necesitan un sistema de archivos en paralelo de alto rendimiento. Los sistemas de archivos compartidos también se pueden crear desde Amazon Elastic File System (EFS) o instancias EC2 con volúmenes de Amazon EBS o volúmenes de almacén de instancias. Con frecuencia, se utiliza un montaje de NFS simple para crear un directorio compartido.

Cuando selecciona su solución de almacenamiento, puede seleccionar una instancia respaldada por EBS para uno o ambos almacenamientos locales y compartidos. Los volúmenes de EBS suelen ser la base de una solución de almacenamiento de HPC. Dispone de distintos tipos de volúmenes de EBS, entre los que se incluyen las unidades de disco duro (HDD) magnéticas, unidades de estado sólido (SSD) de uso general y las SSD de IOPS provisionadas para soluciones de alta IOPS. Se diferencian en el rendimiento, el rendimiento de IOPS y el coste.

Puede obtener aún más mejores de rendimiento si selecciona una instancia optimizada de Amazon EBS. Una instancia optimizada de EBS utiliza una pila de configuración mejorada y ofrece capacidad dedicada adicional para E/S de Amazon EBS. Esta optimización ofrece el mejor rendimiento para sus volúmenes de EBS mediante la minimización de la contención entre la E/S de Amazon EBS y otro tráfico de red hacia y desde su instancia. Elija una instancia optimizada para EBS para obtener un rendimiento más constante y para las aplicaciones de HPC que dependen de una red de baja latencia o tienen necesidades intensivas de datos de E/S para volúmenes de EBS.

Para lanzar una instancia optimizada para EBS, elija un tipo de instancia que permita la optimización de EBS de forma predeterminada o uno que permita la habilitación de la optimización de EBS en el lanzamiento.

Para el almacenamiento temporal a nivel de bloque, se pueden utilizar volúmenes de almacén de instancias, incluidos los volúmenes de SSD (solo disponibles en ciertas familias de instancias) de memoria rápida no volátil (NVMe). Consulte la matriz del tipo de instancia para obtener información sobre la optimización de EBS y la compatibilidad de volúmenes del almacén de instancias.[10]

Cuando seleccione una solución de almacenamiento, asegúrese de que coincida con sus patrones de acceso para lograr el rendimiento deseado. Es sencillo experimentar con diferentes tipos y configuraciones de almacenamiento. Con respecto a las cargas de trabajo de HPC, la opción más costosa no siempre es la solución de mejor rendimiento.

Redes

HPCPERF 4: ¿Cómo selecciona su solución de red?

La solución óptima de red para una carga de trabajo de HPC varía según los requisitos de latencia, ancho de banda y rendimiento. Las aplicaciones de HPC estrechamente acopladas suelen necesitar la menor latencia posible para las conexiones de red entre los nodos de computación. Para cargas de trabajo de tamaño moderado y estrechamente acopladas, es posible seleccionar un mayor tipo de instancia con una gran cantidad de núcleos para que la aplicación se adapte por completo a la instancia sin cruzar la red en absoluto.

Como alternativa, algunas aplicaciones están unidas a la red y necesitan un alto rendimiento de esta. Para estas aplicaciones se pueden seleccionar instancias con un mayor rendimiento de red. El mayor rendimiento de red se obtiene con el tipo de instancia más grande de una familia. Para obtener más detalles, consulte la matriz del tipo de instancia.[7]

Para las aplicaciones grandes y estrechamente acopladas se necesitan varias instancias con baja latencia entre las instancias. En AWS, esto se logra con el lanzamiento de nodos de computación en un grupo de ubicación en clúster, que es un agrupamiento lógico de instancias dentro de una zona de disponibilidad. Un grupo de ubicación en clúster ofrece conectividad sin bloqueo y sin sobresuscripción, que incluye un ancho de banda completo de bisección entre las instancias. Para las aplicaciones estrechamente acopladas sensibles a la latencia que abarcan varias instancias, utilice los grupos de ubicación en clúster.

Además de los grupos de ubicación en clúster, las aplicaciones estrechamente acopladas se benefician de un Elastic Fabric Adapter (EFA), un dispositivo en red que puede asociarse a su instancia de Amazon EC2. EFA ofrece una menor y más constante latencia y un mayor rendimiento que el transporte TCP, que se solía utilizar en sistemas de HPC basados en la nube. Permite un modelo de acceso de derivación del sistema operativo a través de la API de *Libfabric* que permite que las aplicaciones de HPC se comuniquen directamente con el hardware de la interfaz de red. EFA mejora el rendimiento de la comunicación entre

instancias, se optimiza para trabajar en la infraestructura actual de la red de AWS y es fundamental para escalar las aplicaciones estrechamente acopladas.[13]

Si una aplicación no puede beneficiarse de la funcionalidad de derivación del sistema operativo de EFA o un tipo de instancia no es compatible con EFA, se puede obtener un rendimiento óptimo de la red si se selecciona un tipo de instancia que sea compatible con la mejora de las redes. La mejora de las redes ofrece a las instancias EC2 un mayor rendimiento de las redes y un menor uso de la CPU mediante el uso de dispositivos de paso en lugar de dispositivos emulados por hardware. Este método permite que las instancias EC2 logren un mayor ancho de banda, un mayor procesamiento de paquetes por segundo y una menor latencia entre instancias en comparación con la virtualización de dispositivos tradicional.

La mejora de las redes se encuentra disponible en todos los tipos de instancia de esta generación y necesita una AMI con controladores compatibles. Si bien la mayoría de las AMI actuales tiene controladores compatibles, es posible que las AMI personalizadas necesiten controladores actualizados. Para obtener más información sobre la habilitación de la mejora de las redes y la compatibilidad de las instancias, consulte el documento de mejora de las redes.[11]

Por lo general, las cargas de trabajo levemente acopladas no son sensibles a las redes de muy baja latencia y no necesitan usar un grupo de ubicación en clúster ni mantener las instancias en la misma zona de disponibilidad o región.

Revisión
No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas de revisión. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Monitoreo
No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas de monitoreo. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Compensaciones
No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas de monitoreo. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Pilar de optimización de costes

El pilar de **optimización de costes** incluye el proceso continuo de refinamiento y mejora de un sistema de HPC durante todo su ciclo de vida. Desde el diseño inicial de su primera prueba de concepto hasta la operación continua de las cargas de trabajo de producción, la adopción de las prácticas de este documento le permitirá crear y operar sistemas con control de costes

que logren resultados empresariales y minimicen costes. De esta manera, su negocio podrá maximizar el retorno de la inversión.

Principios de diseño

Para HPC en la nube se pueden seguir una serie de principios para lograr la optimización de los costes:

- **Adoptar un modelo de consumo:** solo paga los recursos de computación que consume. Las cargas de trabajo de HPC fluctúan, lo que ofrece la oportunidad de reducir los costes mediante el aumento y la disminución de la capacidad de recursos según sea necesario. Por ejemplo, se puede aprovisionar y reservar por adelantado una capacidad de HPC de bajo nivel de velocidad de ejecución para beneficiarse de mayores descuentos. Por otro lado, se pueden aprovisionar los requisitos de ráfagas con precios al contado o bajo demanda y puestos en línea, solo cuando sea necesario.
- **Optimizar los costes de infraestructura para trabajos específicos:** muchas cargas de trabajo de HPC son parte de una canalización de datos que incluye la transferencia de datos, el preprocesamiento, los cálculos informáticos, el posprocesamiento y los pasos de almacenamiento. La plataforma de informática se optimiza en cada paso, en la nube y no en un servidor grande y costoso. Por ejemplo, si un solo paso en una canalización necesita una gran cantidad de memoria, solo debe pagar un servidor con mayor memoria y más costoso para la aplicación intensiva de la memoria, mientras que todos los demás pasos pueden ejecutarse bien en plataformas de informática más pequeñas y menos costosas. Los costes se reducen con la optimización de la infraestructura para cada paso de una carga de trabajo.
- **Aprovechar las cargas de trabajo de la manera más eficaz**: se obtiene un ahorro en las cargas de trabajo de HPC mediante el ajuste de escala horizontal en la nube. Cuando escala de manera horizontal, muchos trabajos o iteraciones de toda una carga de trabajo se ejecutan en simultáneo para reducir el tiempo total transcurrido. Según la aplicación, la escala horizontal puede ser neutral en cuanto a costes, mientras que ofrece ahorros de costes indirectos por proporcionar resultados en una fracción del tiempo.
- **Utilizar los precios de spot:** las instancia de spot de Amazon EC2 ofrecen más capacidad de cómputo en AWS con grandes descuentos en comparación con las instancias bajo demanda. Sin embargo, se pueden interrumpir las instancias de spot cuando EC2 necesita reclamar la capacidad. Las instancias de spot suelen ser el recurso más rentable para las cargas de trabajo flexibles o tolerantes a fallas. La naturaleza intermitente de las cargas de trabajo de HPC las hace apropiadas para las instancias de spot. Se puede minimizar el riesgo de interrupción de las instancias de spot si se

trabaja con el asesor de spot. Además, se puede mitigar el efecto de la interrupción con el cambio del comportamiento predeterminado de esta y el uso de una flota de spot para administrar sus instancias de spot. La necesidad de reiniciar de manera ocasional una carga de trabajo se compensa con facilidad mediante el ahorro de costes de las instancias de spot.

- **Evaluar la compensación entre coste y tiempo:** las cargas de trabajo estrechamente acopladas y masivamente paralelas pueden ejecutarse en una amplia gama de recuentos de núcleos. Para estas aplicaciones, la eficiencia de ejecución de un caso suele reducirse en los recuentos de núcleos más altos. Se puede crear una curva entre coste y tiempo de respuesta si se ejecutan muchos casos de tipo y tamaño similares. Las curvas son específicas tanto para el tipo de caso como para la aplicación, ya que el ajuste de escala depende en gran medida de la relación entre los requisitos informáticos y los de red. Las cargas de trabajo más grandes pueden escalar aún más que las pequeñas. Si se comprende la compensación entre coste y tiempo de respuesta, las cargas de trabajo urgentes pueden ejecutarse con mayor rapidez y en más núcleos, mientras que se puede lograr un ahorro de costes con menos núcleos y a una máxima eficiencia. Las cargas de trabajo pueden caer en un punto intermedio cuando desea equilibrar la sensibilidad al tiempo y al coste.

Definición

Existen cuatro áreas de prácticas recomendadas para la optimización de costos en la nube:

- Recursos rentables
- Coincidencia de la oferta y la demanda
- Conciencia de gastos
- Optimización a lo largo del tiempo

En el documento técnico del Marco de Buena Arquitectura de AWS se describen las áreas de **coincidencia de la oferta y la demanda**, la **conciencia de gastos** y **la optimización a lo largo del tiempo**.

Prácticas recomendadas

Recursos rentables

HPCCOST 1: ¿Cómo evaluó las opciones disponibles de cómputo y almacenamiento para su carga de trabajo a fin de optimizar costes?

HPCCOST 2: ¿Cómo evaluó las compensaciones entre el tiempo y el coste de la finalización del trabajo?

El uso de las instancias, los recursos y las características adecuadas para su sistema es clave para administrar los costes. Es posible que la selección de la instancia aumente o disminuya el coste total de la ejecución de una carga de trabajo de HPC. Por ejemplo, una carga de trabajo de HPC estrechamente acoplada puede demorar cinco horas en ejecutarse en un clúster de varios servidores más pequeños, mientras que un clúster de menos servidores y más grandes puede costar el doble por hora, pero calcula el resultado en una hora, lo que ahorra dinero en general. La selección del almacenamiento también puede afectar los costes. Considere la posible compensación entre la respuesta del trabajo y la optimización de costes, y pruebe las cargas de trabajo con diferentes tamaños de instancia y opciones de almacenamiento para optimizar los costes.

AWS ofrece una variedad de opciones de precios rentables y flexibles para adquirir instancias de EC2 y otros servicios de la manera que mejor se adapte a sus necesidades. Las instancias bajo demanda permiten pagar la capacidad de cómputo por hora, sin que exista una tarifa mínima necesaria. Las instancias reservadas permiten reservar capacidad y ofrecer ahorros en relación con los precios bajo demanda. Con las instancias de spot, puede aprovechar la capacidad sin utilizar de Amazon EC2 y ofrecer más ahorros en relación con los precios bajo demanda.

Un sistema con buena arquitectura utiliza los recursos más rentables. Además, puede reducir los costes mediante el uso de servicios administrados para el preprocesamiento y el posprocesamiento. Por ejemplo, en lugar de mantener servidores para almacenar y posprocesar los datos de la ejecución completada, estos se pueden almacenar en Amazon S3 y, luego, posprocesar con Amazon EMR o AWS Batch.

Muchos servicios de AWS ofrecen características que reducen aún más sus costes. Por ejemplo, Auto Scaling se integra con EC2 para lanzar y terminar de manera automática instancias basadas en la demanda de carga de trabajo. FSx for Lustre se integra de manera nativa con S3 y presenta todos los contenidos de un bucket de S3 como un sistema de archivos de Lustre. De esta manera, puede optimizar sus costes de almacenamiento mediante el aprovisionamiento de un sistema de archivos de Lustre mínimo para su carga de trabajo inmediata, a la vez que mantiene sus datos a largo plazo en un almacenamiento de S3 rentable. S3 ofrece diferentes clases de almacenamiento para que pueda utilizar la clase más rentable para sus datos. Las clases de almacenamiento Glacier o Glacier Deep permiten archivar datos al costo más bajo.

Experimentar con diferentes tipos de instancia, requisitos de almacenamiento y arquitecturas puede minimizar los costes y mantener el rendimiento deseado.

Coincidencia de la oferta y la demanda
No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas de coincidencia de la oferta y la demanda. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Conciencia de gastos
No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas de conciencia de gastos. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Optimización a lo largo del tiempo
No hay prácticas recomendadas únicas de HPC para el área de prácticas recomendadas de optimización a lo largo del tiempo. Consulte la sección correspondiente en el documento técnico del Marco de Buena Arquitectura de AWS.

Conclusión

Este enfoque ofrece las prácticas recomendadas de arquitectura para diseñar y operar sistemas confiables, seguros, eficientes y rentables para cargas de trabajo de informática de alto rendimiento en la nube. Abarcamos las arquitecturas prototípicas de HPC y los principios generales de diseño de HPC. Analizamos los cinco pilares de buena arquitectura a través del enfoque de HPC y le brindamos una serie de preguntas para ayudarlo a revisar una arquitectura de HPC existente o propuesta. La aplicación del marco a su arquitectura lo ayuda a crear sistemas estables y eficientes, lo que le permite concentrarse en la ejecución de aplicaciones de HPC y en ampliar los límites de su campo.

Colaboradores

Las siguientes personas y organizaciones contribuyeron a redactar este documento:

- Aaron Bucher, arquitecto especialista en soluciones de HPC, Amazon Web Services
- Omar Shorbaji, arquitecto de soluciones globales, Amazon Web Services
- Linda Hedges, ingeniera de aplicaciones de HPC, Amazon Web Services
- Nina Vogl, arquitecta especialista en soluciones de HPC, Amazon Web Services
- Sean Smith, ingeniero de desarrollo de software de HPC, Amazon Web Services
- Kevin Jorissen, arquitecto de soluciones del clima y el tiempo, Amazon Web Services
- Philip Fitzsimons, director senior de buena arquitectura, Amazon Web Services

Documentación adicional

Para obtener información adicional, consulte lo siguiente:

- Marco de Buena Arquitectura de AWS[12]
- https://aws.amazon.com/hpc
- https://d1.awsstatic.com/whitepapers/Intro_to_HPC_on_AWS.pdf
- https://d1.awsstatic.com/whitepapers/optimizing-electronic-design-automation-eda-workflows-on-aws.pdf
- https://aws.amazon.com/blogs/compute/real-world-aws-scalability/

Revisiones del documento

Fecha	Descripción
Diciembre de 2019	Actualizaciones menores
Noviembre de 2018	Actualizaciones menores
Noviembre de 2017	Publicación original

Notes

[1] https://aws.amazon.com/well-architected

[2] https://d0.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf

[3] https://aws.amazon.com/batch/

[4] https://aws.amazon.com/ec2/

[5] https://aws.amazon.com/ec2/spot/

[6] https://aws.amazon.com/message-queue

[7] https://aws.amazon.com/ec2/instance-types/#instance-type-matrix

[8] https://aws.amazon.com/intel/

[9] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/processor_state_control.html

[10] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSOptimized.html#ebs-optimization-support

[11] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html

[12] https://aws.amazon.com/well-architected

[13] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa.html