[ENGLISH]

Hello Cloud Gurus, and welcome to this lecture. And in this lecture,

we're going to cover DynamoDB Accelerator,

and it's also known as DAX. So let's get started.

So what is DAX? Well, DynamoDB Accelerator, or DAX,

is a fully managed, clustered, in-memory cache for DynamoDB.

And it really gives you a massive performance improvement for your DynamoDB

tables, but only for read performance.

And they claim that it can deliver up to a 10x read performance.

That really is significant. So it gives you microsecond performance,

and for millions of requests per second.

So it's ideal for read-heavy and also bursty read workloads,

for example, auction applications, gaming applications,

and retail sites specifically during black Friday promotions or at

Christmas time, or any time when they have a special promotion on.

So for example,

just think of a retail website and they're running a big promotion.

They've got all their product catalog data in DynamoDB. Of course,

while those promotions are on and while there's really good bargains to be had,

a lot of people are going to be wanting to look at their website,

look at all the information about their products,

and decide which products they're going to buy. So in that situation,

DynamoDB would get really hit really hard. There'll be many,

many read operations.

And that's a really good example of when you could use DAX to improve the read

performance for all of those different calls. So how does it work?

Well DAX is a write-through caching service,

and this means that data is written to the cache as well as the backend store at

the same time.

So that means that any time the DynamoDB table gets updated,

you add a new item, or you modify an item.

As well as writing that change into DynamoDB,

it also gets written to DAX as well.

It also allows you to point your API calls at the DAX cluster.

So instead of having your application query DynamoDB,

it will try and query the DAX cluster first.

And if the item you're looking for is in the cache, and that's a cache hit,

DAX will then return the result to the application.

However, if the item is not available, and that's a cache miss,

then DAX performs an eventually consistent GetItem operation against DynamoDB.

So it gets the item out of DynamoDB, it writes into its cache,

and it also hands it back to the application.

So retrieval of data from DAX reduces the read load on your DynamoDB tables.

And in some cases,

you may even be able to reduce the provisioned read capacity on your

tables. So that means you can save some money on your DynamoDB bill. But

what is it not suitable for? Well,

it caters for eventually consistent reads only.

So it's not going to be suitable for applications that require strongly

consistent reads.

So if your application really requires strongly consistent reads,

DAX is not going to be a benefit in that situation.

It's also not really suitable for write-intensive applications.

So you're not going to get a benefit from using DAX because it only helps with

read operations. Also,

applications that don't perform that many read operations are

not really going to see a benefit from configuring DAX,

and the same goes for applications that don't require microsecond response

times. There's no point in configuring it if you don't need that low latency.

So what are our exam tips? Well,

just remember DAX provides in-memory caching for your DynamoDB tables.

It improves the response times for eventually consistent reads only.

When a write operation occurs,

data is written to the DynamoDB Accelerator cache and

the DynamoDB abackend store at the same time.

You point your API calls at the DAX cluster instead of your table.

If the item you're querying is on the cache, DAX will return it. Otherwise,

it's going to perform an eventually consistent GetItem operation to your

DynamoDB table.

And it is not suitable for write-intensive applications,

or applications that require strongly consistent reads.

So that's the end of this lecture. If you have any questions,

please let me know. If not, please feel free to move on to the next lecture.

Thank you.

[SPANISH]

Hola Cloud Gurus, y bienvenidos a esta conferencia. Y en esta conferencia,

vamos a cubrir DynamoDB Accelerator,

y también se conoce como DAX. Entonces empecemos.

Entonces, ¿qué es DAX? Bueno, DynamoDB Accelerator o DAX,

es una memoria caché en memoria completamente administrada y agrupada para DynamoDB .

Y realmente le brinda una mejora masiva en el rendimiento de su DynamoDB .

tablas , pero solo para el rendimiento de lectura.

Y afirman que puede ofrecer un rendimiento de lectura de hasta 10x.

Eso es realmente significativo. Así que te da un rendimiento de microsegundos,

y para millones de solicitudes por segundo.

Por lo tanto, es ideal para cargas de trabajo de lectura intensa y también de lectura en ráfagas ,

por ejemplo, aplicaciones de subastas, aplicaciones de juegos,

y sitios minoristas específicamente durante las promociones del viernes negro o en

Navidad, o en cualquier momento en que tengan una promoción especial.

Así por ejemplo,

solo piense en un sitio web minorista y están ejecutando una gran promoción.

Tienen todos los datos de su catálogo de productos en DynamoDB . Por supuesto,

mientras esas promociones estén activas y mientras haya muy buenas ofertas,

mucha gente va a querer ver su sitio web,

mira toda la información sobre sus productos,

y decidir qué productos van a comprar. Así que en esa situación,

DynamoDB se vería muy afectado. Habrá muchos,

muchas operaciones de lectura.

Y ese es un muy buen ejemplo de cuándo podría usar DAX para mejorar la lectura

rendimiento para todas esas llamadas diferentes. ¿Entonces, cómo funciona?

Bueno, DAX es un servicio de almacenamiento en caché de escritura simultánea,

y esto significa que los datos se escriben en el caché, así como en el almacén de back-end en

al mismo tiempo

Eso significa que cada vez que se actualiza la tabla de DynamoDB ,

agrega un nuevo elemento o modifica un elemento.

Además de escribir ese cambio en DynamoDB ,

también se escribe en DAX.

También le permite apuntar sus llamadas API al clúster DAX.

Entonces, en lugar de que su aplicación consulte DynamoDB ,

intentará consultar el clúster DAX.

Y si el elemento que está buscando está en el caché, y eso es un acierto en el caché,

Luego, DAX devolverá el resultado a la aplicación.

Sin embargo, si el artículo no está disponible, y eso es una falta de caché,

luego , DAX realiza una operación GetItem eventualmente consistente contra DynamoDB .

Así que saca el elemento de DynamoDB , lo escribe en su caché,

y también lo devuelve a la aplicación.

Por lo tanto, la recuperación de datos de DAX reduce la carga de lectura en sus tablas de DynamoDB .

Y en algunos casos,

puede reducir la capacidad de lectura aprovisionada en su

mesas \_ Eso significa que puede ahorrar algo de dinero en su factura de DynamoDB . Pero

qué no es adecuado? Bien,

atiende solo lecturas eventualmente consistentes.

Por lo tanto, no será adecuado para aplicaciones que requieren mucho

lecturas consistentes .

Entonces, si su aplicación realmente requiere lecturas muy consistentes,

DAX no va a ser un beneficio en esa situación.

Tampoco es realmente adecuado para aplicaciones de escritura intensiva.

Por lo tanto, no se beneficiará del uso de DAX porque solo ayuda con

de lectura . También,

las aplicaciones que no realizan tantas operaciones de lectura son

no voy a ver un beneficio al configurar DAX,

y lo mismo ocurre con las aplicaciones que no requieren una respuesta de microsegundos

veces \_ No tiene sentido configurarlo si no necesita esa baja latencia.

Entonces, ¿cuáles son nuestros consejos para el examen? Bien,

solo recuerde que DAX proporciona almacenamiento en caché en memoria para sus tablas de DynamoDB .

Mejora los tiempos de respuesta solo para lecturas eventualmente consistentes.

Cuando se produce una operación de escritura,

los datos se escriben en la caché del Acelerador de DynamoDB y

el DynamoDB una tienda backend al mismo tiempo.

Dirige sus llamadas API al clúster DAX en lugar de a su tabla.

Si el elemento que está consultando está en el caché, DAX lo devolverá. De lo contrario,

va a realizar una operación GetItem eventualmente coherente con su

de DynamoDB .

Y no es adecuado para aplicaciones de escritura intensiva,

o aplicaciones que requieren lecturas fuertemente consistentes.

Así que ese es el final de esta conferencia. Si tienes alguna pregunta,

por favor hágamelo saber Si no es así, siéntase libre de pasar a la siguiente lección.

Gracias.