TITAN

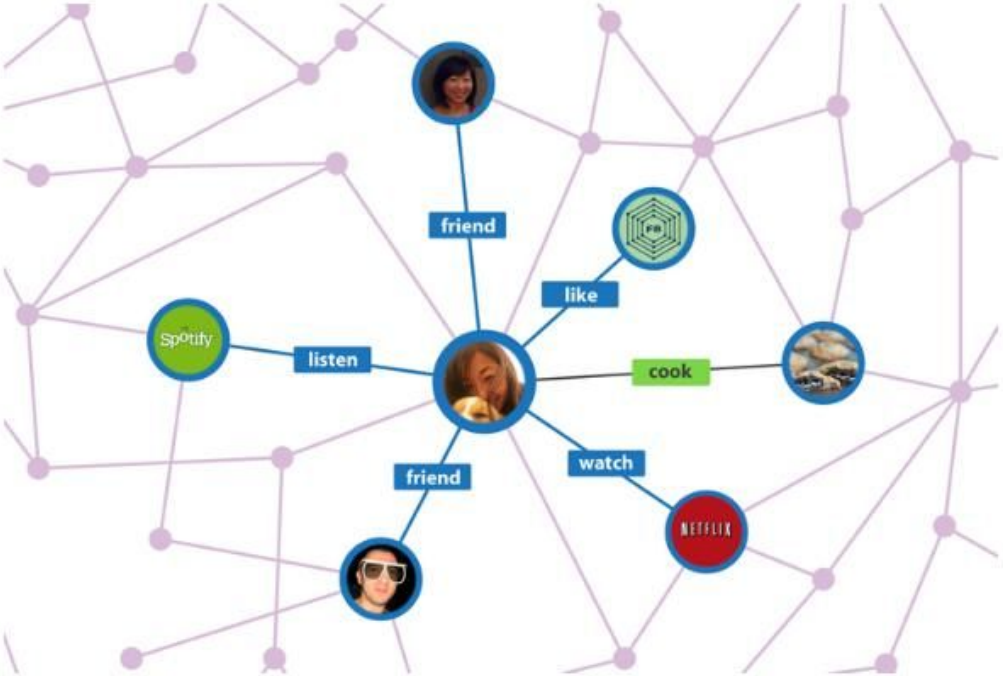
Distributed Graph Database

¿Qué es Titan?

- Base de datos de **grafos**
- **Millones** de vértices
- Base de datos **transaccional**
- Miles de usuarios **concurrentes**

TITAN

Base de datos de grafos



- Información como **nodo.**
- **Relaciones** como aristas.
- Se puede aplicar la **teoría de grafos.**

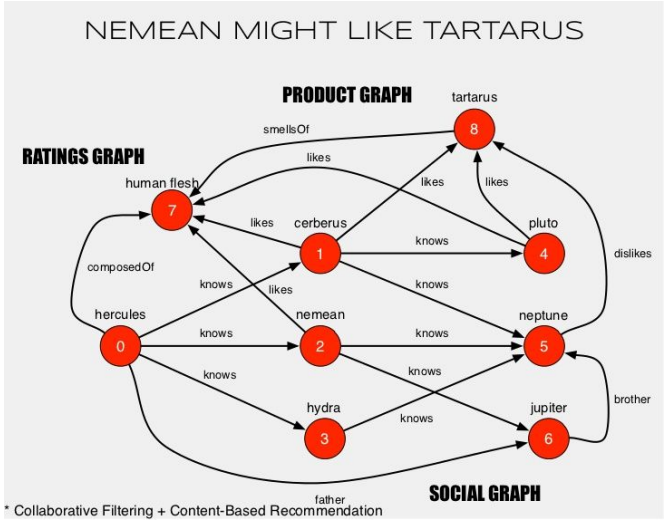
Base de datos de grafos - Ventajas

- Número **indeterminado** de atributos
- Fácil de **recorrer**
- Diferentes **usos** según las relaciones

Query	SQL	Gremlin

Join	`select users.* from users`
	`  inner join groups`
	`  on users.gId = groups.id`
	`  where groups.name = "devs"`
		`g.V('type', 'groups')`
		`  .has('name', 'dev')`
		`  .in('inGroup').map()`

Join-on-join-on-join …	`SELECT TOP (5) [t14].[ProductName]`
	`  FROM (SELECT COUNT(*) AS [value],`
	`  [t11].[ProductName]`
	`  FROM [customers] AS [t8]`
	`  CROSS APPLY (SELECT [t9].[ProductName]`
	`  FROM [orders] AS [t1]`
	`  CROSS JOIN [order details] AS [t2]`
	`  INNER JOIN [products] AS [t3]`
	`  ON [t3].[ProductID] = [t2].[ProductID]`
	`  CROSS JOIN [order details] AS [t4]`
	`  INNER JOIN [orders] AS [t5]`
	`  ON [t5].[OrderID] = [t4].[OrderID]`
	`  LEFT JOIN [customers] AS [t6]`
	`  ON [t6].[CustomerID] = [t5].[CustomerID]`
	`  CROSS JOIN ([orders] AS [t7]`
	`  CROSS JOIN [order details] AS [t8]`
	`  INNER JOIN [products] AS [t9]`
	`  ON [t9].[ProductID] = [t8].[ProductID])`
	`  WHERE NOT EXISTS(SELECT NULL AS [EMPTY]`
	`  FROM [orders] AS [t10]`
	`  CROSS JOIN [order details] AS [t11]`
	`  INNER JOIN [products] AS [t12]`
	`  ON [t12].[ProductID] = [t11].[ProductID]`
	`  WHERE [t9].[ProductID] = [t12].[ProductID]`
	`  AND [t10].[CustomerID] = [t9].[CustomerID]`
	`  AND [t11].[OrderID] = [t10].[OrderID])`
	`  AND [t6].[CustomerID] <> [t9].[CustomerID]`
	`  AND [t1].[CustomerID] = [t8].[CustomerID]`
	`  AND [t2].[OrderID] = [t1].[OrderID]`
	`  AND [t4].[ProductID] = [t3].[ProductID]`
	`  AND [t7].[CustomerID] = [t6].[CustomerID]`
	`  AND [t8].[OrderID] = [t7].[OrderID]) AS [t13]`
	`  WHERE [t6].[CustomerID] = N'ALFKI'`
	`  GROUP BY [t13].[ProductName]) AS [t14]`
	`ORDER BY [t14].[value] DESC`
		`g.V('customerId','ALFKI')`
		`  .as('customer')`
		`  .out('ordered')`
		`  .out('contains')`
		`  .out('is')`
		`  .as('products')`
		`  .in('is')`
		`  .in('contains')`
		`  .in('ordered')`
		`  .except('customer')`
		`  .out('ordered')`
		`  .out('contains')`
		`  .out('is')`
		`  .except('products')`
		`.groupCount().cap()`
		`.orderMap(T.decr[0..<5]`
		`.productName`

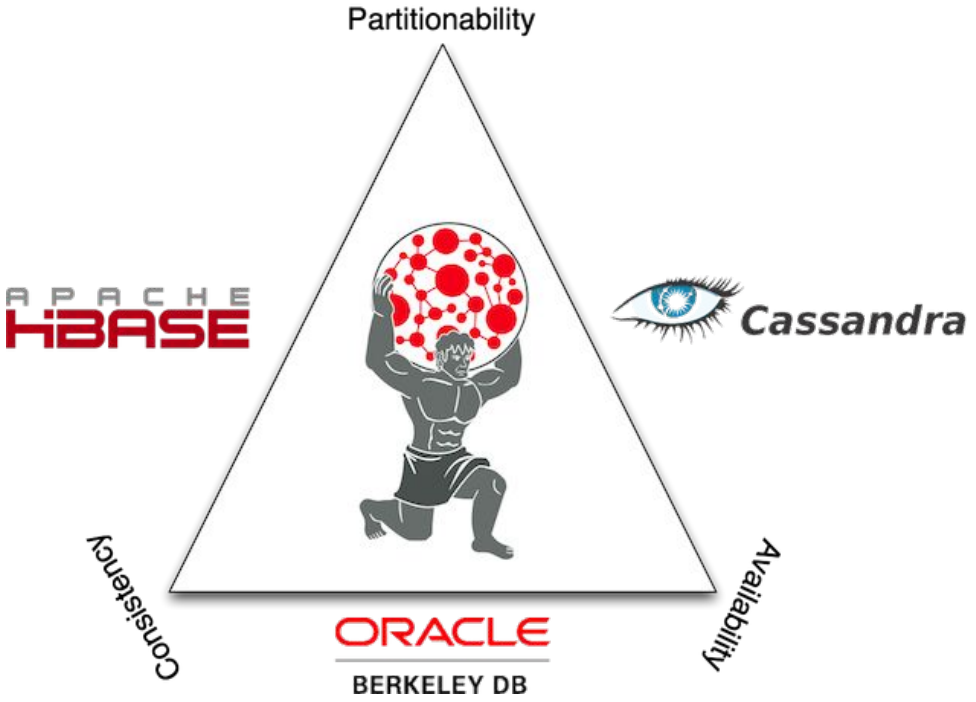


Titan - beneficios

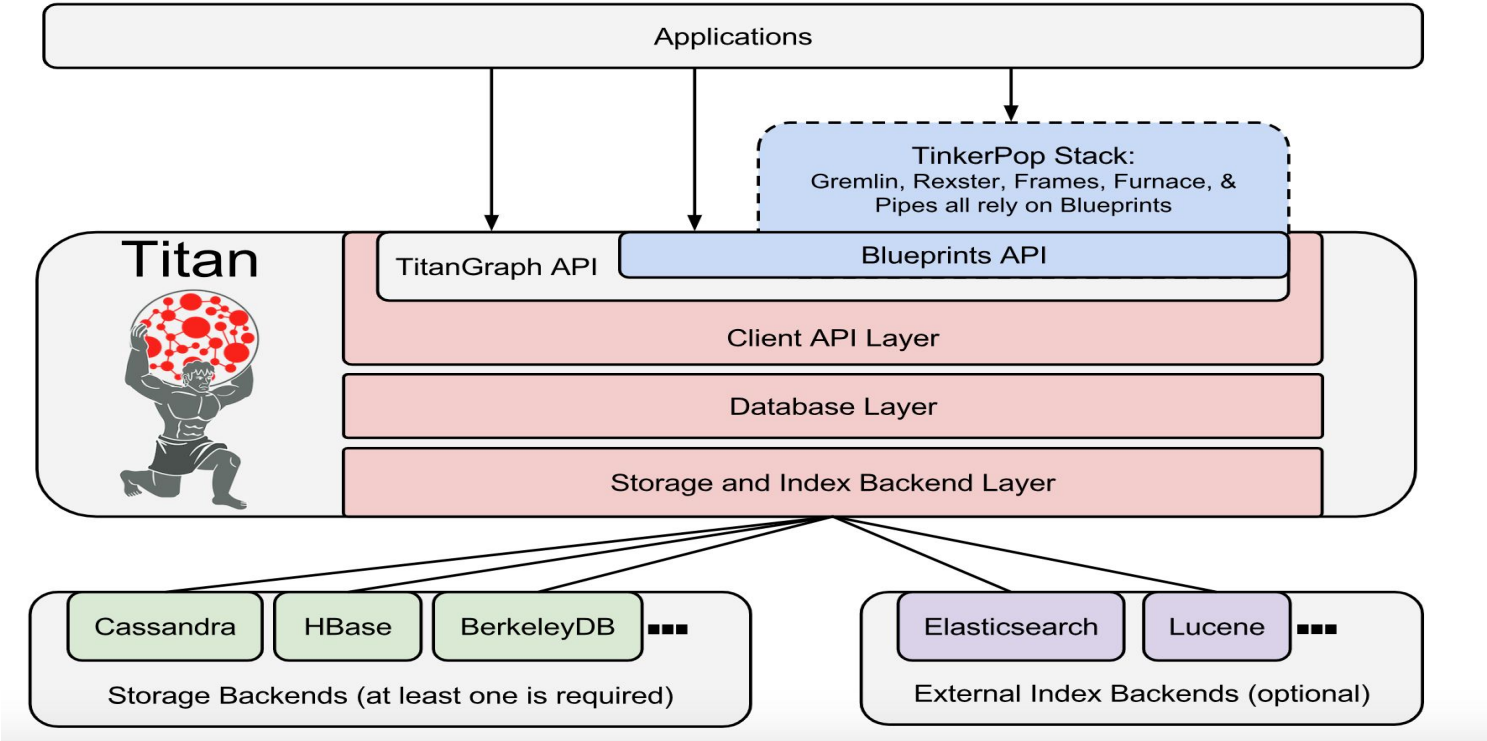
- Los grafos de Titan **escalan** con el número de máquinas en el clúster
- Compatibilidad con análisis de grafos globales y procesamiento de grafos por lotes a través del marco de **Hadoop**.
- Soporte nativo para el popular modelo de datos de gráficos de propiedades expuesto por **TinkerPop.**
- Soporte nativo para el lenguaje de navegación de grafos de **Gremlin.**
- Se ajusta al **teorema CAP.**

Titan - Teorema CAP

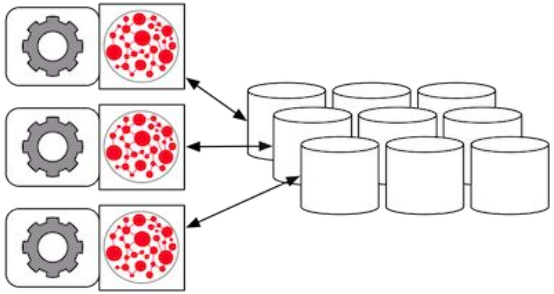
- Titan y Cassandra
  - Continuamente disponible
  - Sin cuellos de botella
- Titan y HBase
  - Estrictamente consistente
  - Continuamente disponible
- Titan y Berkeley DB
  - *Testing*
  - fines de exploración



Titan - Arquitectura

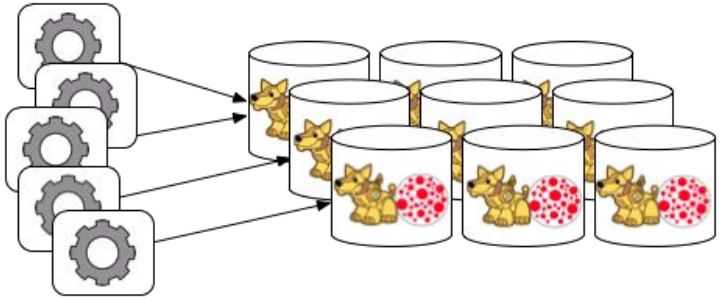


Titan - Arquitectura 2

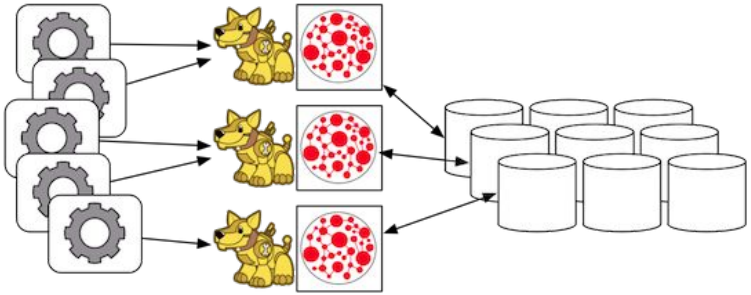


Nativo

Remoto



Embebido



Titan - Limitaciones

- Espacio posible

Titan soporta un trillón (2^60) de aristas y la mitad en vértices. Esta limitación la impone el esquema de ID que ocupa Titan.

- Búsqueda

Obtener una arista por su ID no tiene un tiempo de operación constante es O(log (k)) donde k es el número de bordes incidentes en el vértice adyacente.

Is TitanDB dead? #1360

Closed **vlad-alexandru-ionescu** opened this issue on Nov 9, 2016 · 5 comments

vlad-alexandru-... commented on Nov 9, 2016

I noticed that the documentation and home page of Titan now redirect to a closed, enterprise-only version provided by DataStax. Also, there haven't been commits to this repo since June. Am I correct to assume that the open-source version of Titan is no longer being maintained?

With both the author of Titan and the author of Gremlin now being employees of DataStax it looks like there's little interest in maintaining a free and open community for Graph databases in general, let alone Titan. This kind of monopoly is what kills innovation and has historically killed great products like CouchDB/CouchBase.

What's going on?

👍 8

manigandham commented on Feb 18, 2017 • edited

@vlad-alexandru-ionescu @mamoit

TitanDB is retired, there is a new project called **JanusGraph** which is now part of the Linux foundation. It starts with the foundations of TitanDB and has more interest and backing then before.

http://janusgraph.org/

https://opensource.googleblog.com/2017/01/janusgraph-connects-past-and-future-of-titan.html

👍 30

JanusGraph

Distributed graph database

Docs • GitHub • Download

JanusGraph is a scalable graph database optimized for storing and querying graphs containing hundreds of billions of vertices and edges distributed across a multi-machine cluster. JanusGraph is a transactional database that can support thousands of concurrent users executing complex graph traversals in real time.

DEMO

TITAN

Distributed Graph Database