Target

1. How graph databases are compared traditional SQL databases and other types of NoSQL databases?
2. How to perform graph analytics on top of Titan/Neo4j graph database?
3. How is the scalability of Titan/Neo4j graph database and analytics on top of the database.

Titan graph database

1. Project website: <http://titan.thinkaurelius.com/>
2. Titan Documentation: <http://s3.thinkaurelius.com/docs/titan/1.0.0/>
3. [Gremlin\_(programming\_language)](https://en.wikipedia.org/wiki/Gremlin_(programming_language)) for Titan.
4. Apache TinkerPop graph computing framework which include Gremlin, Titan and many others: <http://tinkerpop.apache.org/>
5. <http://tinkerpop.apache.org/docs/current/reference/#neo4j-gremlin>
6. <http://tinkerpop.apache.org/docs/current/reference/#hadoop-gremlin>
7. <http://tinkerpop.apache.org/docs/current/reference/#interacting-with-spark>
8. My code to save csv graph file to Titan

|  |
| --- |
| graph = TinkerGraph.open()  g = graph.traversal()  new File('../interferon-beta-network.csv').eachLine {  if (it.startsWith("\"")){  line = it.split(',')  def fromVertex  def toVertex  if (!g.V().has('name', line[0])) {  fromVertex = graph.addVertex('name', line[0], 'type', line[1], 'id', line[2]) }  else {  fromVertex = graph.vertices().findAll{ it.property('name').value().equalsIgnoreCase(line[0])}[0] }  if (!g.V().has('name', line[3])) {  toVertex = graph.addVertex('name', line[3], 'type', line[4], 'id', line[5]) }  else {  toVertex = graph.vertices().findAll{ it.property('name').value().equalsIgnoreCase(line[3])}[0] }  fromVertex.addEdge('link', toVertex)  }  }  g.V().values("name") |

Neo4j graph database

<https://neo4j.com/>