Apache S2Graph(incubating) At Kakao
A Large Scale Distributed Graph Database
1. Technical Challenges.
2. What is S2Graph.
3. Use Cases.
4. Real World Use Cases.
Technical Challenges
Why We Got into Graph Databases

Graph Databases
- Relationship-Oriented
- Expressive yet Simple

Our Data
- Highly Connected
- Complex Relations

+30 services at Kakao are powered by S2Graph
Social Graph?

Relations + Activities = Social Graph

- 🐻 Muzi is friend of 🐻 Ryan

- 🐻 Muzi listen to “Hello” by Adele.

- 🐻 Ryan is playing KakaoGame
Our Social Graph

Friend

- View count
- Write length
- Play level
- View Content
- Advertise
- Listen count
- Coupon price
- Emoticon
- Like count
- Eat rating
- Comment
- Search keyword
- Group size
- Present price
- Affinity 1
- Affinity 2
- Affinity 3
- Affinity 4
- Affinity 5
- Affinity 6
- Affinity 7
- Affinity 8
- Affinity 9

Keyword: Present price: 3
Technical Challenges

1. Large graph constantly changing.
   1. Total # of Edge: + 1 trillion and growing.
   2. Social Network: more than 10 billion edges, 200 million vertices, 50 million realtime update on social network.
   3. User activities (Click, Like, Share, Buy): 2.5 ~ 3 billion real-time incoming edges, 50 billion batch processed edges.

2. Low latency for breadth first search traversal on connected data.
   1. Peak graph-traversing query per minute: 4 million
   2. Average response time: 50 ms

3. Update should be applied into query result in real-time.

4. Support for Dynamic Ranking logic
   1. push strategy: hard to change data ranking logic dynamically.
   2. pull strategy: can try various data ranking logic
Why Did We Build It?

Existing solutions weren’t performant enough for our needs.

Especially,

1. Maintaining a mutable graph at scale was not supported. (i.e. Updates/ Deletes were slow!)
2. Breadth First Search traversal was not fast enough.

So we built our own!
Before

Each app server should know each DB’s sharding logic.
Highly inter-connected architecture
After

S2Graph DB
stateless app servers

Apache HBase

Messaging App
SNS App
Blog App
What is S2Graph?
What is S2Graph?

- Property Graph Model: Vertices + Edges + Properties
- S2Graph = Property Graph Model + Scalability + Fast CRUD Operations
- Graph-processing layer atop HBase
Example: News Feed (cont)

```
SELECT a.*, b.*
FROM    friends a, user_posts b
WHERE a.user_id = b.user_id
      AND b.updated_at >= yesterday
      AND b.action_type in ('create', 'like', 'share')
```

Posts that my friends interacted.

```
curl -XPOST localhost:9000/graphs/getEdges -H 'Content-Type: Application/json' -d '```
Example: Recommendation (User-based CF) (cont)

SELECT a.*, b.*
FROM similar_users a, user_products b
WHERE a.sim_user_id = b.user_id
AND b.updated_at >= yesterday

Products that similar user interact recently.

curl -X POST localhost:9000/graphs/getEdges -H 'Content-Type: Application/json' -d '{
  "filterOut": {
    "srcVertices": [{"serviceName": "s2graph", "columnName": "user_id", "id": 1},
      "steps": [{"label": "user_products_interact"}]
  },
  "srcVertices": [{"serviceName": "s2graph", "columnName": "user_id", "id": 1},
    "steps": [
      {"label": "similar_users", "direction": "out", "limit": 100, "where": "similarity > 0.2"}, // step
      {"label": "user_products_interact", "direction": "out", "limit": 10,
        "where": "created_at >= yesterday and price >= 1000"}
    ]
  }
}'}
Example: Recommendation(Item-based CF) (cont)

**user-product interaction**
(click/buy/like/share)

```
SELECT a.*, b.*
FROM similar_a, user_products b
WHERE a.sim_user_id = b.user_id
AND b.updated_at >= yesterday
```

**Similar Products**

Products that are similar to what I have interested.

```
curl -X POST localhost:9000/graphs/getEdges -H 'Content-Type: Application/json' -d 
{
    "srcVertices": [{"serviceName": "s2graph", "columnName": "user_id", "id":1}],
    "steps": [
        [{"label": "user_products_interact", "direction": "out", "limit": 100,
          "where": "created_at >= yesterday and price >= 1000"}],
        [{"label": "similar_products", "direction": "out", "limit": 10, "where": "similarity > 0.2"}]
    ]
}
```
Example: Recommendation (Spreading Activation) (cont)

```
SELECT b.product_id, count(*)
FROM user_products a, user_products b
WHERE a.user_id = 1
AND a.product_id = b.product_id
GROUP BY b.product_id
```

Products that is interacted by users who interacted on products that I interact

```bash
curl -XPOST localhost:9000/graphs/getEdges -H 'Content-Type: Application/json' -d '{
  "srcVertices": [{"serviceName": "s2graph", "columnName": "user_id", "id":1}],
  "steps": [
    {"label": "user_products_interact", "direction": "out", "limit": 100, "where": "created_at >= yesterday and price >= 1000"},
    {"label": "user_products_interact", "direction": "in", "limit": 10, "where": "created_at >= today"},
    {"label": "user_products_interact", "direction": "out", "limit": 10, "where": "created_at >= 1 hour ago"}
  ]
}'
```
Use Cases
1. Storage for user activities and relationships

Friends, clicks, purchases, likes, shares, comments, etc.
2. Real-time Recommendation (Spreading Activation)

1. Find items a user has reacted to (clicked, purchased..)
2. Find other users who reacted to the same items.
3. Find other items that those users reacted to.

Figure 5: Real-time collaborative filtering by Spreading Activation
2. Real-time Recommendation (cont)

Composite multiple queries via weighted sum

EX) Ensemble of recommendation algorithms

1. Item-based collaborative filtering (CF)
2. User-based CF
3. Matrix Factorization
4. Demographical MP items (for cold-start problem)
3. Native A/B (Bucket) Testing

1. Bucket = Query
2. Track performance of each bucket realtime: CTR, conversion rate.
Real World Use Cases
1. User Based Collaborative Filtering
2. Item Based Collaborative Filtering.
3. Matrix Factorization.
4. Content Based Collaborative Filtering.
5. Most Popular.
7. Spreading Activation.
8. Social Recommendation.

Weighted sum of each algorithms are also possible.
1. Emoticon Store
Baseline (most popular) vs S2Graph

- CTR: +137.52%
- Purchase/Click: +70.8%
- Purchase/Impression: +304%

Before -> After

- Total Item Click: x3.8
- Unique User with Item Click: x3.5
2. Kakao Talk Channel: Recommend Card, Social Card
Baseline (most popular) vs S2Graph

Total Click: +123%
CTR: +36.6%
Unique # of contents with clicks: x20
2. Kakao Talk Channel: Social Card (cont)

Baseline vs S2Graph

Total Click: +591%
CTR: +23.65%
3. Kakao Talk Gift Shop
3. Kakao Talk Gift Shop (cont)

Baseline (most popular) vs S2Graph

CTR: +26.03%
Purchase/Click: +344%
Purchase/Impression: +459%
Powered By S2Graph

Social Platform
- KakaoTalk
- KakaoStory
- KakaoGroup
- Plain
- Brunch
- Path

Contents Platform
- Media Daum
- KakaoGame
- Emoticon Store
- KakaoPage
- KakaoMusic
- Daum tvPot

Commerce Platform
- Kakao TV
- 1 boon

Marketing Platform
- Gift Shop
- KakaoStyle
- Story Plus

Local Platform
- KakaoPlace

Advertise Platform
- Daum DDN
Statistics

1. total # of edges(activities + relations) >= 1 trillion
2. daily, new incoming edges in realtime >= 3 billion
3. daily, new edges that processed from batch process >= 50 billion
4. average query per minute >= 200 million. peak >= 400 million. under 50 ms.
5. 40% queries are 3 step query, 40% are 2 step, 20% are 1 step.

Operations

1. # of HBase region server = 40
2. # of query server = 70
3. # of write server = 20
Entire projects are Apache Incubator project. This means S2Graph is open to anyone.