Blend in Chicago: MongoDB World 2017

Yubo Su

Blend

June 20, 2017

MONGODB WORLD'17

blend

- Tom Schenk, Chief data officer, Chicago. *WindyGrid*.
  - Track colocated data, 911 calls to Tweets to weather.
  - Flexible schema: `{what, when, where}`
  - Predictive analytics (example, where to send food inspector) using visualization of multiple causal layers.
- Dev Ittycheria, CEO MongoDB
  - 2007 is watershed year, AWS, iPhone, Android, and many others.
  - Argue b/c storage costs dropped below a critical point.
  - MongoDB also in 2007: document model, distributed systems + aggregation.

MONGODB WORLD'17

blend

- Eliot Horowitz, CTO, MongoDB
  - 3.6 ships November, already on Github.
  - MongoDB Charts (3.6)
    - Business Intelligence: BI Connector is SQL interface.
    - Coercing data to table is difficult: polymorphic schemas, arrays.
    - Solution: *MongoDB Charts*! Data visualization tool, handles above.
  - 3.6 document model features:
    - `$lookup` takes sub-pipelines!
    - `$update` can operate on arrays natively! Takes a filter over array entries, can iterate over nested.
    - JSON Schemas.
  - 3.6 distributed systems:
    - Native retryable writes
    - *Change Streams* can get a stream of changes to a db.

- Eliot Horowitz, CTO, MongoDB (continued)
  - Mongo Atlas
    - “Should be irrespensible to run MongoDB in cloud w/o Atlas”
    - Built in security, one-click spin up, built in scaling elasticity.
    - Data browser + performance viewer in UI (utilization stats, examine queries as stream, explore data),
    - Live migration service (not very live in demo, requires downtime for mirror to catch up and change source of truth).
    - Now with MS Azure + Google Cloud support too (+ AWS).

    - Performance Adviser.
    - CRUD support in data browser.
    - Charts!
    - LDAP Auth.
    - Cross-region, cross-cloud!
  - MongoDB Stich (Beta as of today in Atlas, 06/20/17)
    - “Backend as a service”
    - REST API for MongoDB
    - Configuration-based auth/security
    - Service composition to govern how services talk to each other.

MONGODB WORLD'17

blend

- Nested schema, spectrum of highly normalized or denormed storage.
  - Normalized requires foreign keys, requires looking into many collections.
  - Denorm is simpler query, complex schema.
- Consider three possible behaviors:
  - Get player: Denorm outperforms.
  - Add new field to doc: either add new collection or modify every doc, the same.
  - Change existing field: If a highly shared field, normalized is very fast.

- Optimizing highly normalized:
  - Can optimize with aggregate, but more importantly `db.createView()`.
    - Views are basically stored aggregates.
    - Better `$project` support.
  - Also consider, if reading much more than writing, should store calculated fields!
- Optimizing denormed:
  - Should normalize fields that are infrequently updated.
- `tl;dr` normalized have fast write, slow reads. Should embed everything that is infrequently updated.

MONGODB WORLD'17

blend

- Axiom: data models maximize performance + scalability despite latency, costs, hardware.
- Common issues #1, too many optional fields:
  - Use attribute array, `[{key: keyName, value}]`.
  - Accommodates optional fields.
- Common issues #2, working set does not fit in RAM.
  - Can subset, truncate data
  - Probably also useful for showing users too.
- Common issues #3, data consistency.
  - Accept instantaneous inconsistency, duplicate at regular intervals ☹.
- Common issues #4, repeated computations
  - Reads generally outnumber writes, apply computation on write.
- Common issues #5, expensive tracking
  - e.g. expensive to increment on every page view
  - Solution: random number in range $[1, N]$, increment by N.
- Common issues #6, large data easily overflow
  - Bucket, store buckets into a separate collection.

MONGODB WORLD'17

blend

- Microservices vs. monolith, preferable b/c web scale, faster iteration, compartmentalized.
- One common rule of thumb is that one developer can own the whole thing, a couple hundred lines, but not everybody
- Hard metal vs. Docker (Kubernetes) vs. Atlas.
- Kafka can run general events while Mongo streams (the new feature) only handles database updates.

- Query system overview:
  - Input: JSON
  - Parse into tree
  - Generate plan (which indicies for which leaves of the tree)
  - Plan selection: try all of the plans for a trial period, see which one was fastest (Note: plan caching)
  - Execution & return
- `OR`s inside of `AND`s is a pain for plan generation.
  - `AND` is considered indexed when one child has index.
  - `OR` is considered indexed when all children have indicies.
  - `OR`s have to dedupe by hashing to merge the two results.
- Problem: no tight index bounds on these queries
  - *Tight* index bounds are when all documens in index bounds match the query.
  - (As opposed to when a parent node imposes a filter, `FETCH`)
- Bounds are not tight b/c two branches of children cannot talk to each other! e.g. the `OR` will not be tight since the `AND` above will have to further fetch against its other child.

MONGODB WORLD'17

blend

- Solution: Disjunctive Normal Form?
  - `AND` with `OR` child solved!
  - Exponentially many plans though, index choices at each child.
- Solution: OR-pushdowns! Predicates pulled up to the `AND` parent and pushed down into any `OR` children if they can tighten index bounds.
  - Note that this is not imposed as an extra `AND` condition, just metadata for the recursive query planner to plan against.
- Paper: Query Optimization by Predicate Move-Around

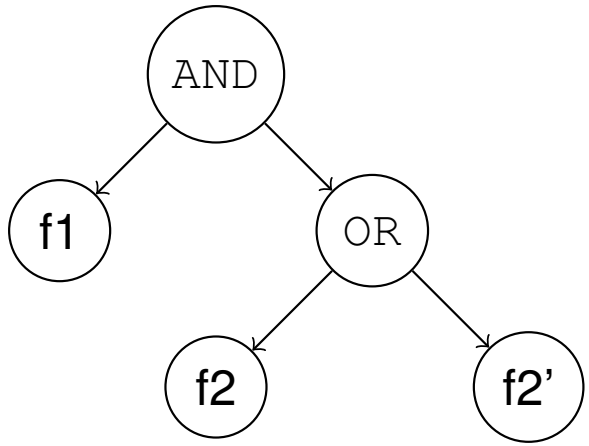

Figure: `AND` with `OR` child. Consider if index is `{f1, f2}`? `{f2, f1}`?

- Key to geograhpic colocation.
- Zone sharding + replica sets
  - Zone sharding: shards per region.
  - Replica sets are `mongod` processes that share the same data.
- Configuration is:
  - One primary in each region, each a separate zone shard.
  - One secondary in its own region (prio 3), two in another (prio 2), and two more in a third (prio 1, 0) for symmetry across all regions, hidden secondary.
  - Spread across multiple regions, odd number of voting members, primary DC members should have higher priorities.

- Can specify region on read/write.
- Upshot is that can do multi-region writes while guaranteeing local availability on read.
- Configurable to write to secondary especially if primary lives in a different region.
- Can configure with `MAX_STALENESS` parameter for when a cluster can be read from.

MONGODB WORLD'17

blend

- Built an object store for internal communications, chat + attachment retention, Mongo backbone.
- RESTful so easy to expose HTTP link as a db.
- Nested schema corresponding to URL: `/users/:id/chat/convs/X/messages/Y`
- Storage (chat), collection (convs), sub-collection (messages), documentIds

- Pubsub (user is online) can be done by consuming the oplog on the db primary.
- To shard and hide it to the user, just need some lookups userId $\rightarrow$ sharding keys.

MONGODB WORLD'17

blend

Evening Keynotes

06/20/17 1715–1845

- Saska Mojsilovic, IBM
  - Need for more data in health for precision health service distribution.
  - All sorts of orgs estimating and predicting from sparse data.
- Claudio Gosiker (Florida Blue) & Alan Chhabra (MongoDB)
  - Use data for healthcare outreach, personalizable views for customer reps.
- Matt Parker, Stand-up Mathematician!

MONGODB WORLD'17

blend