EXPLORING DATOMIC

PUG/IP - 2014.02.10
JAMES POLERA [@UNCRYPTIC] [HTTP://BITZERO.ORG]
JAMES@UNCRYPTIC.COM

About Me

- My name is James Polera, I'm the IT manager for a mid-size law firm in Union county, and part of the sister technology company (they do .NET primarily). I also own a consulting company where I do contract work in my spare time.

- I've been working in IT professionally since 2000 in both Sysadmin and Developer roles (often at the SAME TIME).

- I’ve been participating as an organizer for PUG/IP since late 2012.

- I like to call myself an autodidactic polyglot.

ON TO DATOMIC

What isn't Datomic?

- Open Source
- SQL
- NoSQL
- Update in place

What is Datomic?

- Fact storage
- Not just that - accretive fact storage
- “NewSQL”
- Free*
- Commercially supported by Cognitect (Pro version)

Datomic Components

- Peer library
- Transactor
- Storage (memory, local, advanced)

The Peer Library

- Is embedded in your application
- Provides: data access, querying and caching
- Talks to the transactor
- Speaks Datalog (more on that later)

The Peer Library

- Is JVM based (supported on Java and Clojure)
- Is accessible via a REST API for non-JVM languages
- *Really* tricky to get working with Jython (also buggy due to an apparent implementation issue in Jython) - https://groups.google.com/forum/#!msg/datomic/b8iODxDmmvM/OowdRu6oQ2wJ

The Transactor

- Manages transactions and changes to your data
- Your writes go through here
- Your deletions (or retractions) go through here

The Storage Service

- Available for Pro only
- Can be:
  - Dynamo DB
  - Riak
  - Couchbase
  - Infinispan
  - SQL

Feature matrix

	Datomic Free	Datomic Pro Starter	Datomic Pro
Embedded Datalog Engine	Yes	Yes	Yes
Embedded Memory Database	Yes	Yes	Yes
Transactor-local Storage	Yes	Yes	Yes
Acid Transactions	Yes	Yes	Yes
Advanced Storage Service	No	Yes	Yes
Datomic Console	Yes	Yes	Yes
High Availability Transactor	No	Yes	Yes
Integrated memcached	No	Yes	Yes
Number of Processes	2 peers + transactor	2 peers + transactor	Licensed Limit
Redistributable	Yes	No	No (OEM Available)
License	Free	Free (with registration)	Paid, perpetual
Support	Community	Community	Included w/maintenance

Data from http://www.datomic.com/pricing.html

LET’S TALK DATA

But first, some basics: Datom

- Entity **James**
- Attribute **Likes**
- Value **Coffee**
- Transaction **2014-02-10T18:30:00**

But first, some basics: Datom

- Datoms are immutable
- Datomic has no concept of “UPDATE Users SET Username=‘somenewname’ where UserId=42;”
- At this point in time UserId 42 has ‘somename’
- If that changes, Datomic retains the old data, but moves the new data to the top.

But first, some basics: Datalog

- It's older than Datomic (Datomic has an implementation of Datalog)
- Has roots in Prolog
- It's embedded
- …which means your queries happen locally

Datalog

- Datomic's Datalog deals in sets of tuples of the form [Entity Attribute Value Transaction]
- http://www.learndatalogtoday.org/

OK, REALLY
LET'S TALK DATA

Extensible Data Notion (EDN)

- Schemas are defined using EDN
- Sample schema: tvdb-schema.edn

LET'S CONNECT TO DATOMIC

The Transactor

- Start it up from your Datomic directory with “bin/transactor config/samples/dev-transactor-template.properties”

The Java Shell

- Start it up from your Datomic directory with “bin/shell”
- Let’s setup our database.

The Datomic Console

- Start it up from your Datomic directory with “bin/console -p <PORT> <ALIAS> <datomic_connection_string>”
- i.e. “bin/console -p 8899 free “datomic:free//localhost:4334/“
- Point your web browser to http://localhost:8899/browse

Anatomy of a Datalog Query

- Let's get all of our shows (should only be 1 at this point)

```
[:find ?shows
:where [?s :show/name ?shows]]
```

Anatomy of a Datalog Query

- What about passing parameters?

```
[:find ?shows
 :in $ ?name
 :where [?p :person/name ?name]
        [?s :show/actors ?p]
        [?s :show/name ?shows]]
```

Anatomy of a Datalog Query

- What about filtering in the query? (with a little bit of Clojure)

```
[:find ?shows
 :in $ ?name
 :where
 [?p :person/name ?name]
 [?s :show/actors ?p]
 [?s :show/name ?shows]
 [(.startsWith ?shows "D")]]
```

Accessing from Python

- Using the REST interface
- Start it up with: bin/rest -p 8080 free “datomic:free://localhost:4334/“
- https://github.com/tony-landis/datomic-py
- To the IPython Notebook!

Resources

- http://www.learndatalogtoday.org/
- https://github.com/edn-format/edn
- http://docs.datomic.com/
- http://www.infoq.com/articles/Architecture-Datomic
- https://github.com/tony-landis/datomic-py

Resources

- My Datomic playlist - http://www.youtube.com/playlist?list=PLazWoTyz1FjpswrAVKyTenxq9ALO4Yg0y
- https://github.com/polera/exploring_datomic

THANK YOU