

# HBASE DATA TYPES (WIP)

Nick Dimiduk

Member of Technical Staff, HBase

HBase Birds of a Feather, Hadoop Summit 2013



# MOTIVATION

- `byte[ ]` is necessary but not sufficient for building applications
- every application must (re-)solve it's own subset of the "serialization problem"
- ecosystem tools must agree in order to interoperate
- application code full of pre/post client API-call serialization litter



# DESIGN

1. think holistically -- not just Java
2. encoding format that preserves non-encoded order
3. user-extensible API for defining types
4. provide basic data types out of the box
5. extend client API with type support
6. extend MapReduce API with type support



# TRICKY BITS

- encoding format => data on disk => painful to change
- respect encoding performance overhead in tight loops
- user extensible, user usable API
  - realistic API that real apps can build against
  - don't impost on POJOs (no forced interfaces, &c)
  - avoid magic (no ASM, no AOP, avoid **ORM scorn**)
- community agreement on what data types to ship out of the box
- more stuff I haven't {thought of,encountered} yet



# INSPIRATION

- order-preserving encoding strategies: [SQLite4](#), [Orderly](#), [Phoenix](#)
- user extensible data types: [PostgreSQL](#)
- (SQL) data types and API supporting execution engines [Hive](#), [Pig](#), [Phoenix](#), [Impala](#)
- Experiments with client APIs: [Typo](#)



# IMPLEMENTATION

- [HBASE-8089](#): Add type support (*parent ticket*)
- [HBASE-8201](#): Implement serialization strategies (*patch available*)
- [HBASE-8694](#): Performance evaluation of serialization strategies (*unassigned*)
- [HBASE-8693](#): Implement extensible type API based on serialization primitives (*WIP*)
- [HBASE-7941](#): Provide client API with support for primitive types (*unassigned*)
- [HBASE-8593](#): Type support in ImportTSV tool (*WIP*)



# FUTURE DIRECTIONS

- consider data types anywhere else users touch HBase
  - bundled MR jobs
  - `Filters`
- extend `RegionServer` API with type support (?!?)
  - Coprocessors?
  - `StoreFile` formats?
- non-Java implementation for non-Java clients



**THANKS!**

