



LLVM Engine

S19 15-721 Final Presentation

Amadou, Katia, Tanuj, Wan
Building on Prashanth's work



Overview / Project Goals - Status Update

- 75% → Extending TPL DSL
 - ✓ Sorting with or without limits
 - ✓ Aggregations
 - ✓ Block + Index Nested Loop Joins
- 100% → Integrate with Terrier
 - ✓ CI checks
 - ✓ Lower Layers (storage, index, WAL, transactions)
 - ✓ Design “push-based data flow” for Plan Node conversion. Stub out classes
 - [WIP] Implementing the stubs.
 - [Note] Had success translating COUNT(*) + SeqScanPlanNode, but refactoring now to more scalable design
- 125% → [Nope] Implementing debug annotations for LLVM opcodes



Overview / Nutshell

- Along the way
 - ✓ Sorter and accompanying tests
 - ✓ Aggregates and accompanying tests
 - ✓ Preliminary format/tidy/etc CI to get ready for merging into terrier
 - ✓ Index Joins : pulled in indexes, index lookup and iterator added.
 - ✓ Storage layer, Transactions, Output to upper layers.
 - [WIP] Snapshot of TPL has been ported to terrier
 - It compiles and runs, but CI blows up.
 - [WIP] Convert Plan Nodes
 - [changes-requested] PR for fixing TPL scope resolution
- Somewhere in between the 75% - 100% goal



Correctness - TPL

- Correctness is tested with a mix of C++ code and TPL code
 - C++ code is for a specific functionality.
 - TPL code is for an end-to-end test.
 - Currently these are all handwritten, eventually we can generate them from plan nodes.



Correctness - Terrier

- Plan nodes were merged last Tuesday, their serialization was merged last Thursday
- Currently: hand-craft expected input JSON and manually inspect `ast::AstDump` of output
- Future: programmatically check AST output

Migrating to Terrier's storage caused about 2x slowdown compared to Prashanth's existing no-transactions "storage".



Code Quality

- Some duplicated logic in common/ and execution/util, this needs to be sorted out
- Decision: should execution continue living in a separate namespace?
- Chasing memory leaks
- Need to double-check all lifetime assumptions
 - `ast::Identifier` does not own its string, no problems so far but still
 - `util::Region` is optimized for small allocations, it deallocates all at once on destruction



Concrete Tasks (aka how we're spending finals week)

- We expect to finish these by the end of the semester / before leaving for break
- Pipeline design and plan node format have been stabilized
 - Bring over the translators for various types of nodes
- Code quality
 - Remove offending duplicate functionality (e.g. macros)
 - Fix certain `#define`'s that conflict across TPL and Terrier, e.g. `#define RESTRICT`
- Adapt tests to plan node JSON examples
 - Replace current `ast::AstDump` usage with programmatic checks

Future Work

- Hooking up OutputBuffer to the network layer / whoever wants execution results

