Project Goal:
For an OLTP workload containing distributed transactions, one would like to predict when transactions are not single-partitioned. We are exploring optimizations for these types of workloads for the H-Store parallel, main-memory DBMS using partitioned predictive Markov models.

Inputs:
- **Stored Procedures**
  H-Store requires that administrators pre-define stored procedures comprised of Java control code intermixed with parameterized SQL statements.
- **Sample Workload Trace**
  The administrator must also provide a sample workload trace for the target application. A trace record consists of (1) the stored procedure executed, (2) the queries executed in the transaction, and (3) the input parameters.

OLTP Markov Models:
Using the workload trace, the modeler generates a separate Markov model for each stored procedure. Each graph is specific to the initial partition in the database that can invoke the stored procedure. The vertices represent the potential execution states of a transaction, including (1) the executed query and (2) what partitions have been read/written up to that point. The edges weights represent the probabilities of one state transitioning to another.

Runtime Optimizations:
An OLTP client sends a transaction request to a single H-Store execution node in the cluster.

The node's transaction coordinator invokes its procedure executor and estimates the execution path of the new transaction.

The estimator tracks state and sends the coordinator the relevant partition ids. The coordinator informs others on whether it is safe to execute new transactions on those partitions before the original transaction commits.

When the transaction commits or aborts, the node recomputes edge weights if the transaction deviated from the original predictions.