**Motivation**
- Reduce tail latency (high-percentile latency) of user queries, e.g., 99th percentile
- Reduce extreme tail latency at each index server, e.g., 99.99th percentile

**Why Extreme Tail Latency?**
- The 99th-percentile response time < 120ms
- The 99.99th-percentile response time < 120ms

**Reduction Tail Latency by Parallelization**
- **Opportunity**
  - Available idle cores
  - CPU-intensive workloads
- **Challenges**
  - 1. Parallelizing all queries (inefficient)
  - 2. Parallelizing short queries (no speed up)

**Parallelize the predicted long queries only**

**DDS (Delayed-Dynamic-Selective) Prediction**
- **Delayed prediction**
  - Complete many short queries sequentially
  - Collect dynamic features
- **Selective prediction**
  - Predictor for confidence level
  - Predictor for execution time

**Delayed prediction**

**Dynamic features**
- Collected at query runtime
  1. NumEstMatchDoc := # current matched docs / # processed docs
  2. Statistics of the dynamic score distribution

**Selective prediction**
- Parallelize the unpredictable queries
- Parallel query if
  - Predicted execution time > α
  - Predicted L₁ error > β

**Importance of dynamic features**
- Top-5 feature importance by boosted regression tree
- NumEstMatchDoc helps to predict # total matched doc
- MinDynScore helps to predict early termination

**Predictor accuracy**
- Baseline: PRED
- 957% precision improvement at 98.9% recall over PRED

**Simulation results on tail latency reduction**
- Baseline S
  - Prediction before running a query
  - Parallelize the long query
- Proposed DDS
  - Run a query for 10ms sequentially
  - Parallelizes the predicted long or unpredictable queries