THE GOAL: Which storage engine and tuning is best for an application?

```
Put((5, -10)), Get(5), Put((7, -17)), Get(10), Get(5), Get(7), Get(10), Put((5, -17))
```

AVERAGE-CASE ANALYSIS: A Distribution-Aware Framework

Workload Feature Vector

<table>
<thead>
<tr>
<th>Distributions</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>Access pattern</td>
</tr>
<tr>
<td>Frequency</td>
<td>put</td>
</tr>
<tr>
<td></td>
<td>range</td>
</tr>
<tr>
<td>Operation type</td>
<td>get</td>
</tr>
</tbody>
</table>

Storage Engine Configuration

- engine = RocksDB
- size ratio = 10
- buffer size = 1GB
- filter memory = 2GB

EXAMPLE: Get Cost on LSM

"Early stopping"

\[ C_{\text{get}} = (1 - C_{\text{buffer}}) \sum_{i=1}^{L} p_i \cdot C_i \]

\[ \mathbb{P}[\text{"false" access at level } i] \]

\[ \mathbb{P}[\text{key in buffer}] \]

\[ \mathbb{P}[\text{query ongoing at level } i] \]

VERIFICATION: Key-Value Stores

Distribution-aware model captures workload I/Os!

USE CASES: How can these models be utilized?

- Compare designs
- Automate selection
- Search design space
- Create self-designing storage engine

10 mil keys; workload (20,000 ops): 50% reads/50% writes