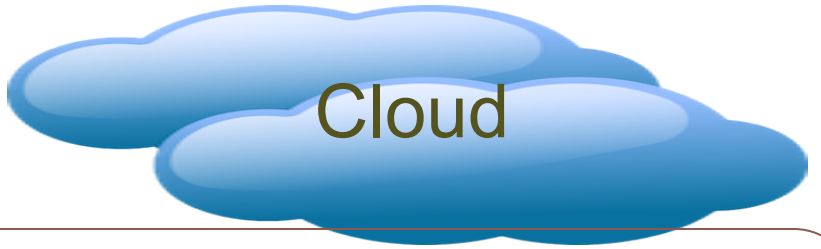
Releasing Cloud Databases from the Chains of Prediction Models

Ryan Marcus and Olga Papaemmanouil
Brandeis University

Cloud Databases Landscape



Database-as-a-Service

- Managed DBMS
- Relational & NoSQL DBs

Microsoft SQL Azure

ORACLE Cloud Database

Amazon RDS

IaaS-based DB Instances

- Non managed DBMS
- Do It Yourself model

Infrastructure as a Service (IaaS)

rackspace the open cloud company

amazon web services

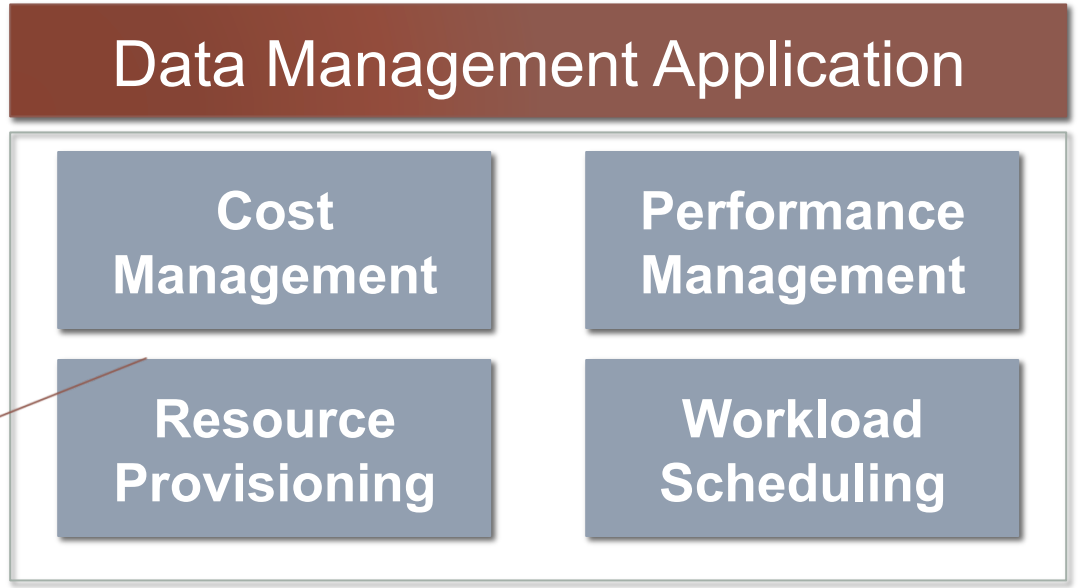
Azure

Google Compute Engine

Deployment Challenges

NP-hard problem

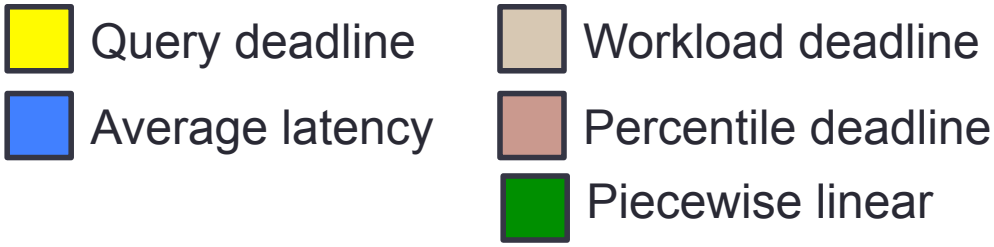
- Automatic scale up & down
- Query routing & scheduling
- Cost vs performance



State-of-the-art

<table>
<tr><th>Placement</th><th colspan="2">Provisioning</th><th>Scheduling</th></tr>
<tr><td>PMAX
(Liu et al.)</td><td>Auto
(Rogers et al.)</td><td>SmartSLA
(Xiong et al.)</td><td>Shepherd
(Chi et al.)</td></tr>
<tr><td colspan="4">SLATree
(Chi et al.)</td></tr>
<tr><td colspan="3">Multi-tenant SLOs
(Lang et al.)</td><td>iCBS
(Chi et al.)</td></tr>
<tr><td>Delphi / Pythia
(Elmore et al.)</td><td colspan="3">Hypergraph
(Çatalyürek et al.)</td></tr>
<tr><td>SCOPE
(Chaiken et al.)</td><td colspan="2">Bazaar
(Jalaparti et al.)</td><td>many traditional methods ...</td></tr>
</table>

State-of-the-art



Placement	Provisioning		Scheduling
PMAX (Liu et al.)	**Auto** (Rogers et al.)	**SmartSLA** (Xiong et al.)	**Shepherd** (Chi et al.)
SLATree (Chi et al.)			
Multi-tenant SLOs (Lang et al.)			**iCBS** (Chi et al.)
Delphi / Pythia (Elmore et al.)	**Hypergraph** (Çatalyürek et al.)		
SCOPE (Chaiken et al.)	**Bazaar** (Jalaparti et al.)		many traditional methods ...

Performance Prediction Models

- DBMS-related challenges
  - isolated vs. concurrent query execution
  - known vs unseen query types (“templates”)
  - extensive off-line training
  - state-of-the-art: 15-20% prediction error
- Cloud-related challenges
  - numerous resource configurations
  - dynamic environment: “noisy neighbors”

Wish List

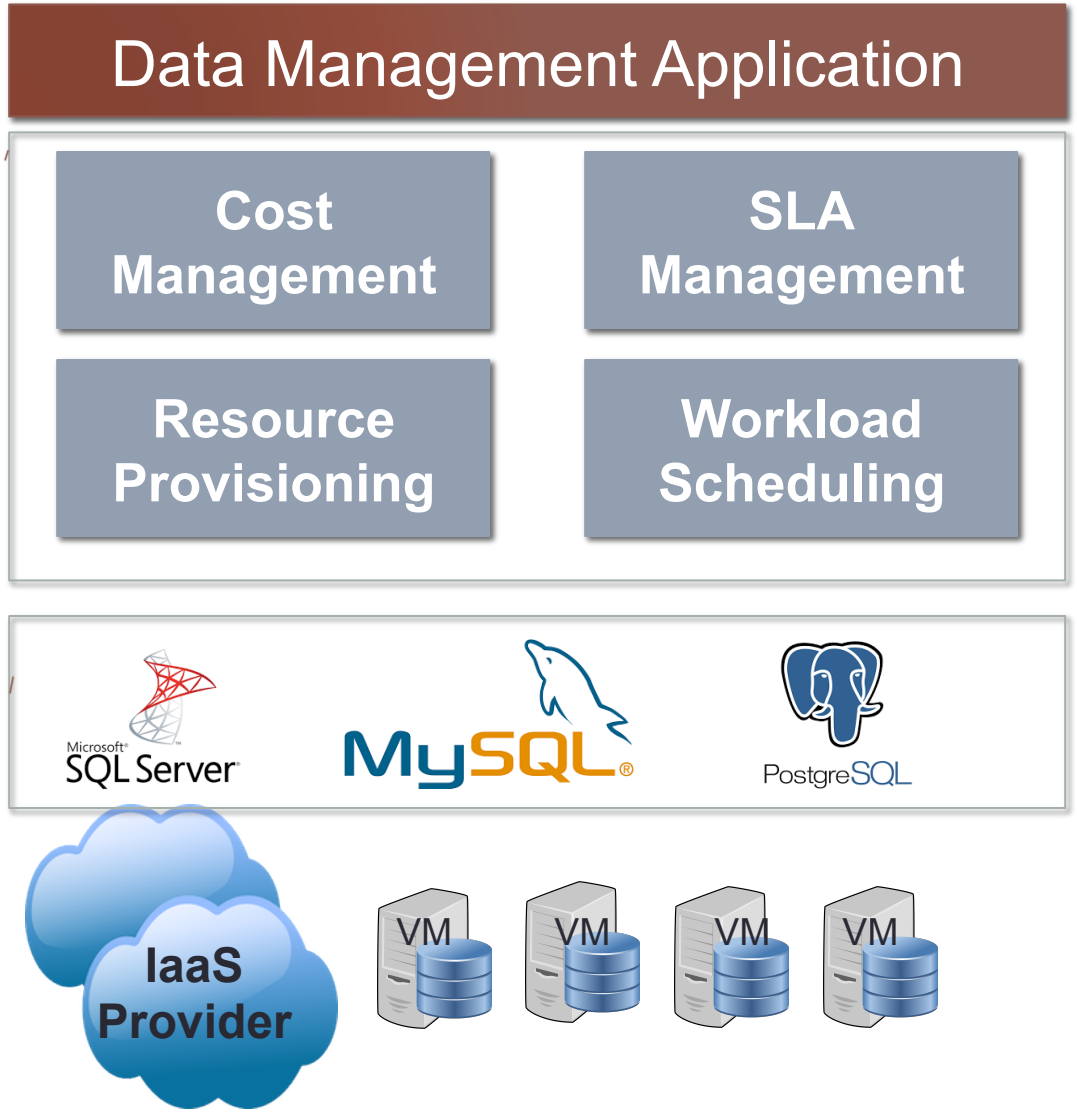
	Challenges
End-to-end cost-aware service (resource provisioning, workload scheduling)	**complex interactions**
Application-defined performance goals (per query deadline, percentile, average latency, max latency)	**arbitrary goals**
Agnostic to workload characteristics (templates, arrival rates, execution times)	**arbitrary workloads**
Dynamic resource availability	**arbitrary resources**

ML approach: model dynamic, complex decisions

Bandit: ML-Based Cost Management

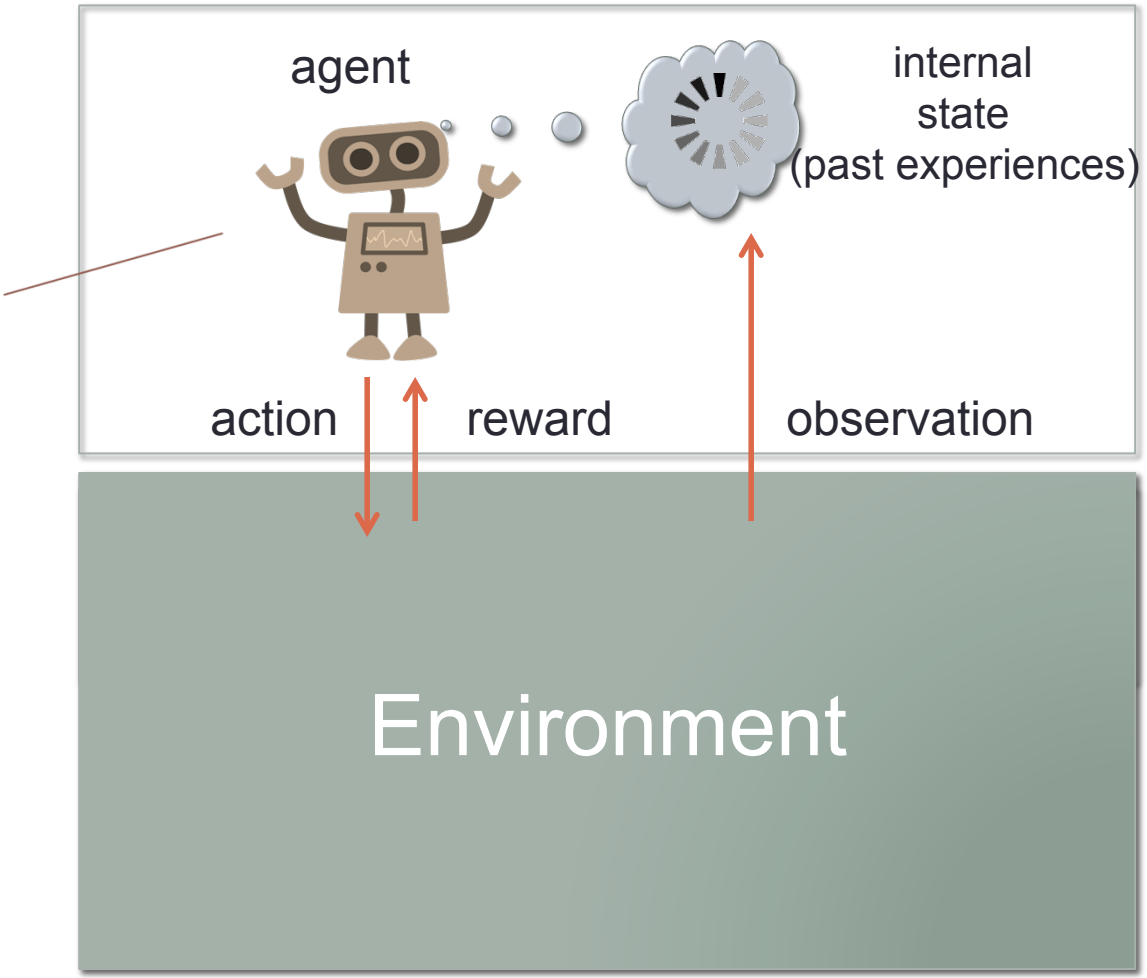
- **Reinforcement learning**
- **Online** workload distribution & resource provisioning

- OLAP Read only workloads
- **Performance goals (SLAs)**
- **Minimize $$$$**
  resource usage fees
  +
  SLA violations fees



Reinforcement Learning

- Continuous learning
- Explicit reward modeling
- Action selection
  - maximize reward

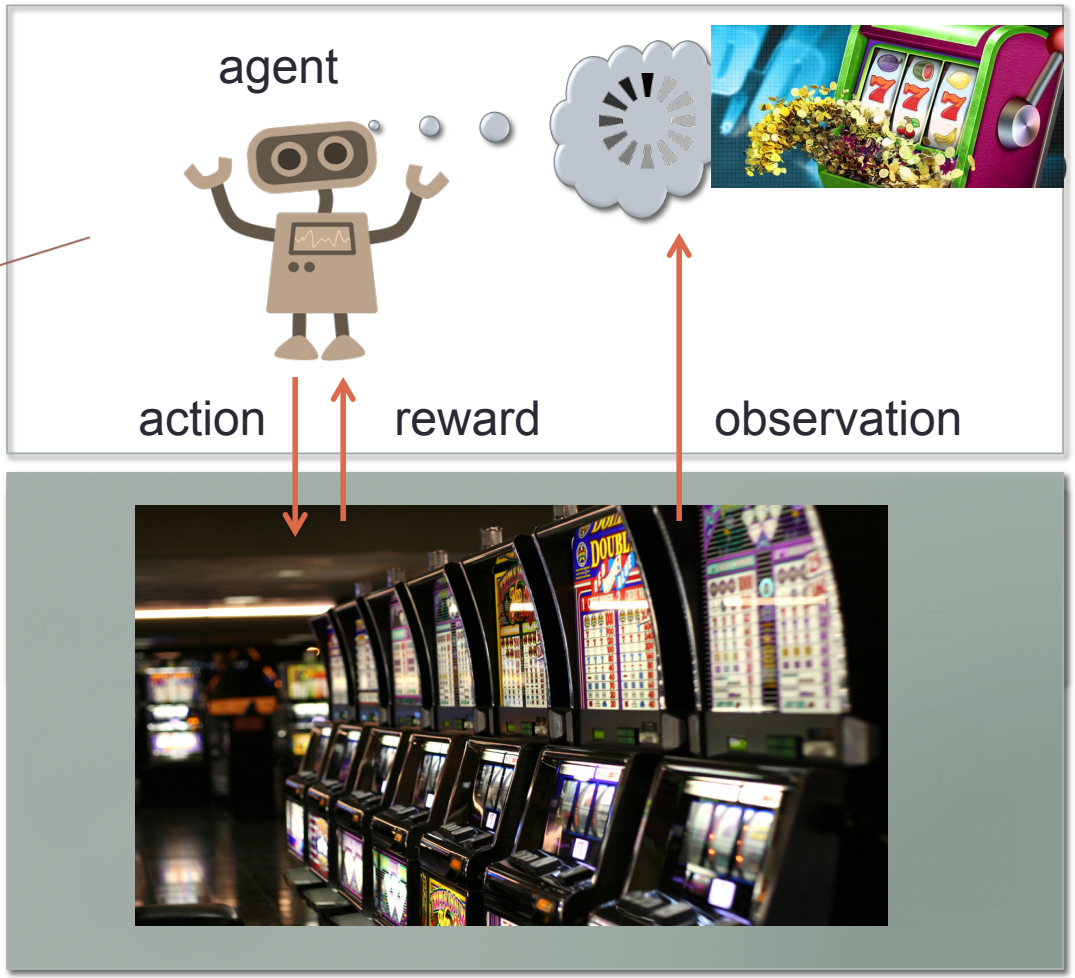


CMABs
(Contextual Multi-Armed Bandits)

Contextual Multi-Armed Bandit Problem

Armed Bandit = Slot Machine

*Which slot machine to play (**action**) so that you walk out with the most $$$ (**reward**)?*



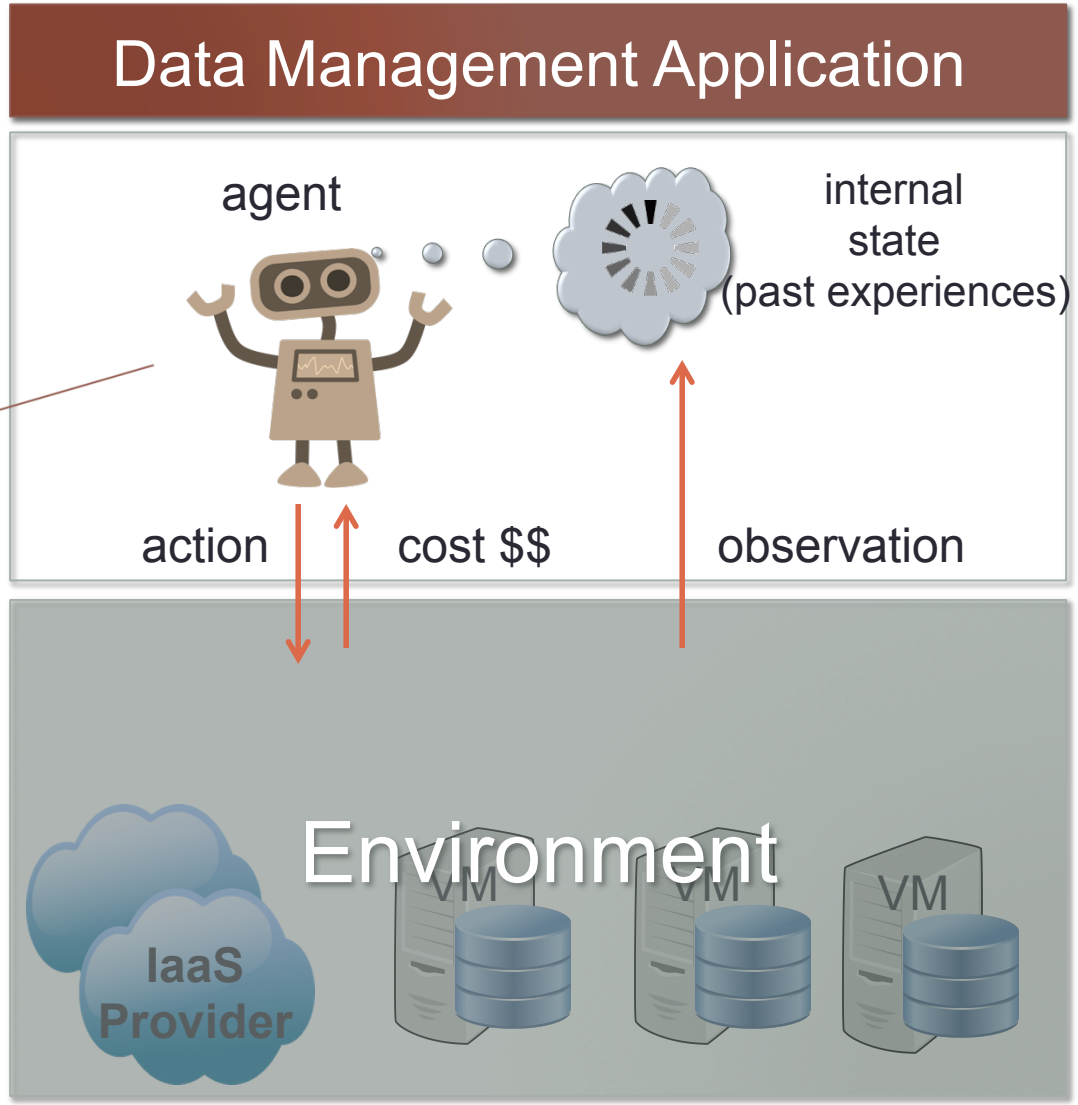
CMABs in Bandit

(Contextual Multi-Armed Bandits)

Contextual Multi-Armed Bandit Problem

Slot Machine = Virtual Machine

*Which machine to use (new/ old) (**action**) so that you execute the incoming query with minimum cost $$ (**cost**)?*



CMABs in Bandit

(Contextual Multi-Armed Bandits)

Action (per VM)

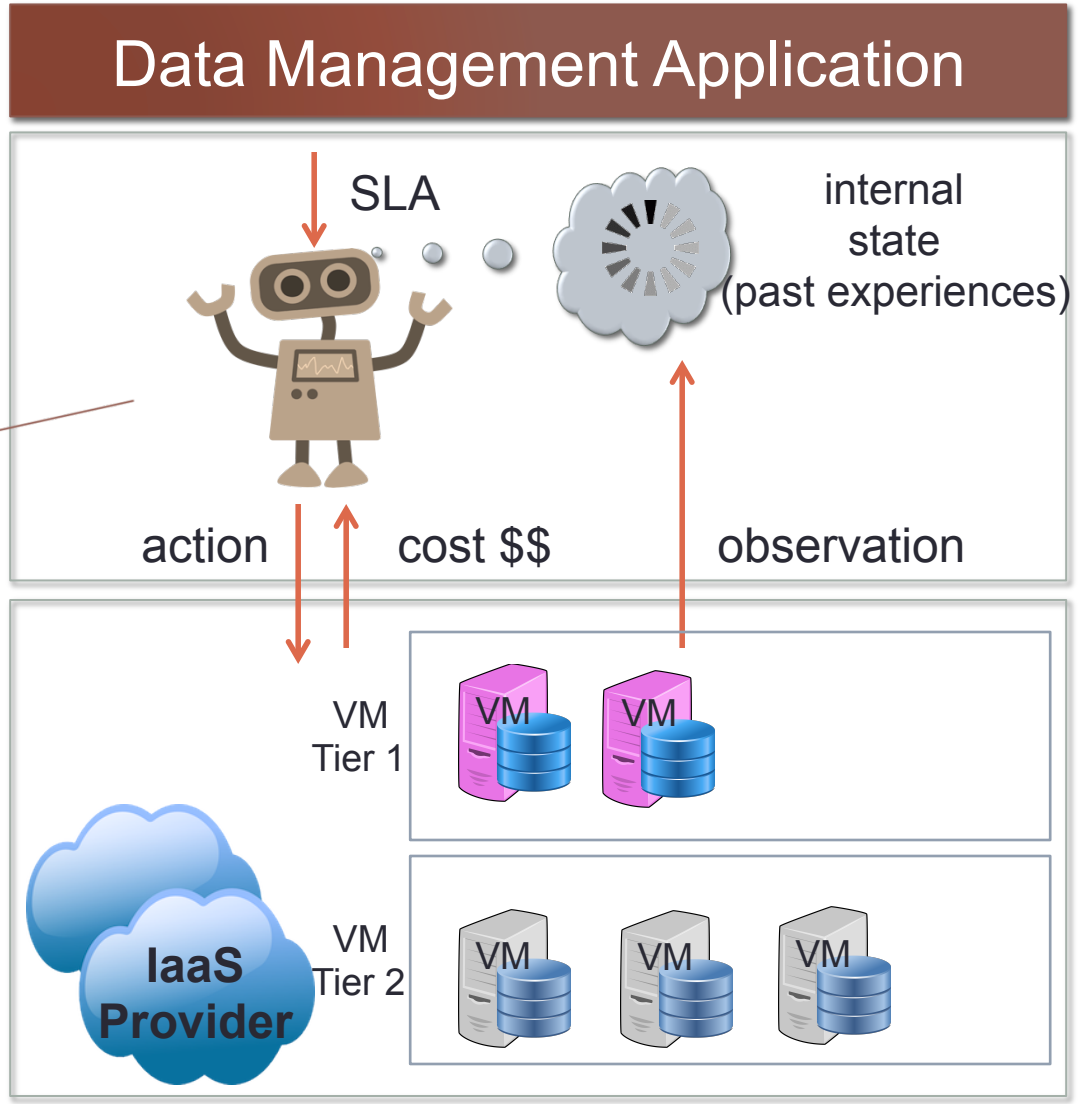
- Accept
- Pass to next /new VM
- Down one VM tier

Reward

- $$ cost: processing & SLA violation penalties

Observation

- context of the decision
- action
- $$ cost



CMABs in Bandit

(Contextual Multi-Armed Bandits)

Action (per VM)

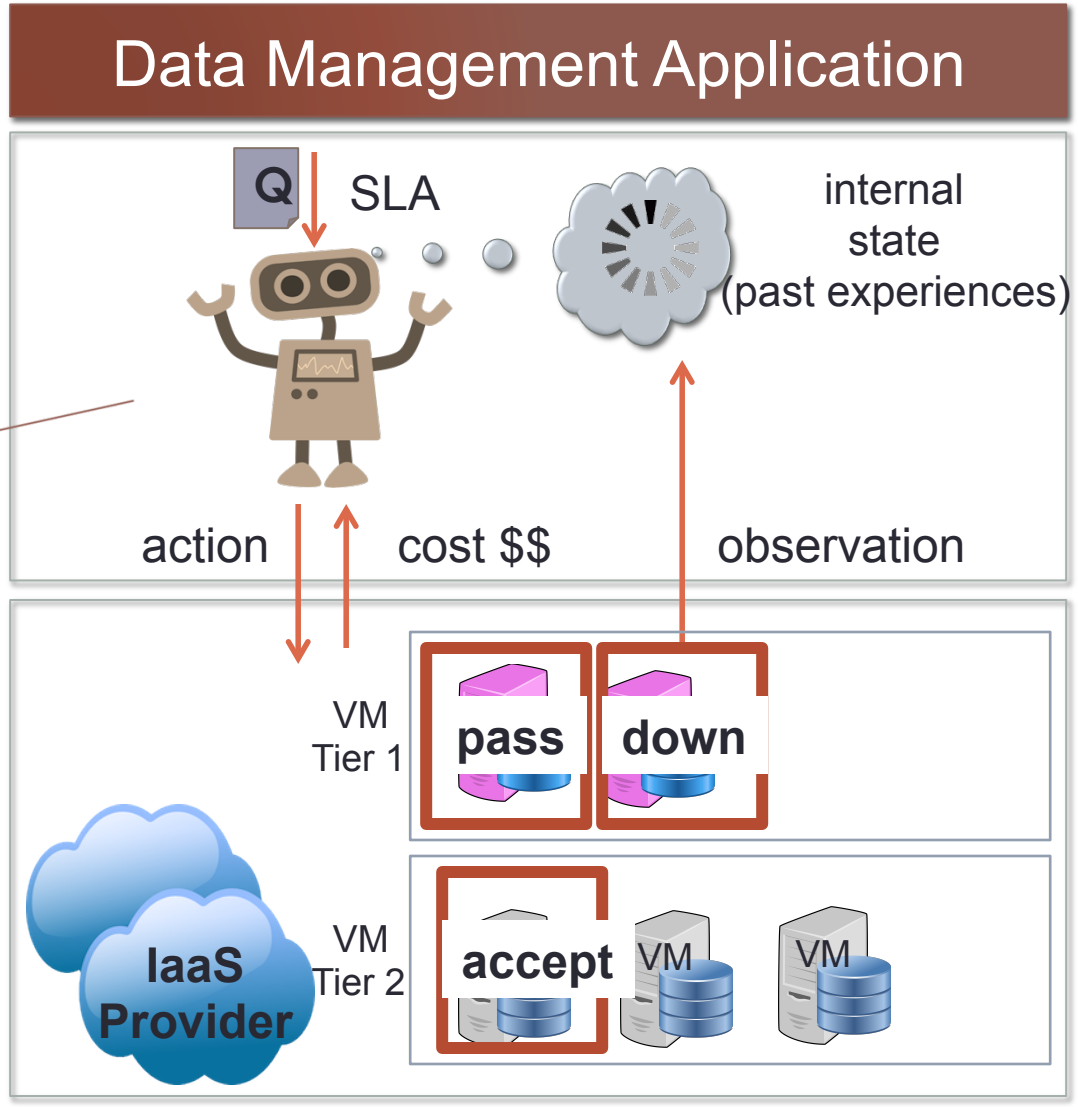
- Accept
- Pass to next /new VM
- Down one VM tier

Reward

- $$ cost: processing & SLA violation penalties

Observation

- context of the decision
- action
- $$ cost



CMABs in Bandit

(Contextual Multi-Armed Bandits)

Action (per VM)

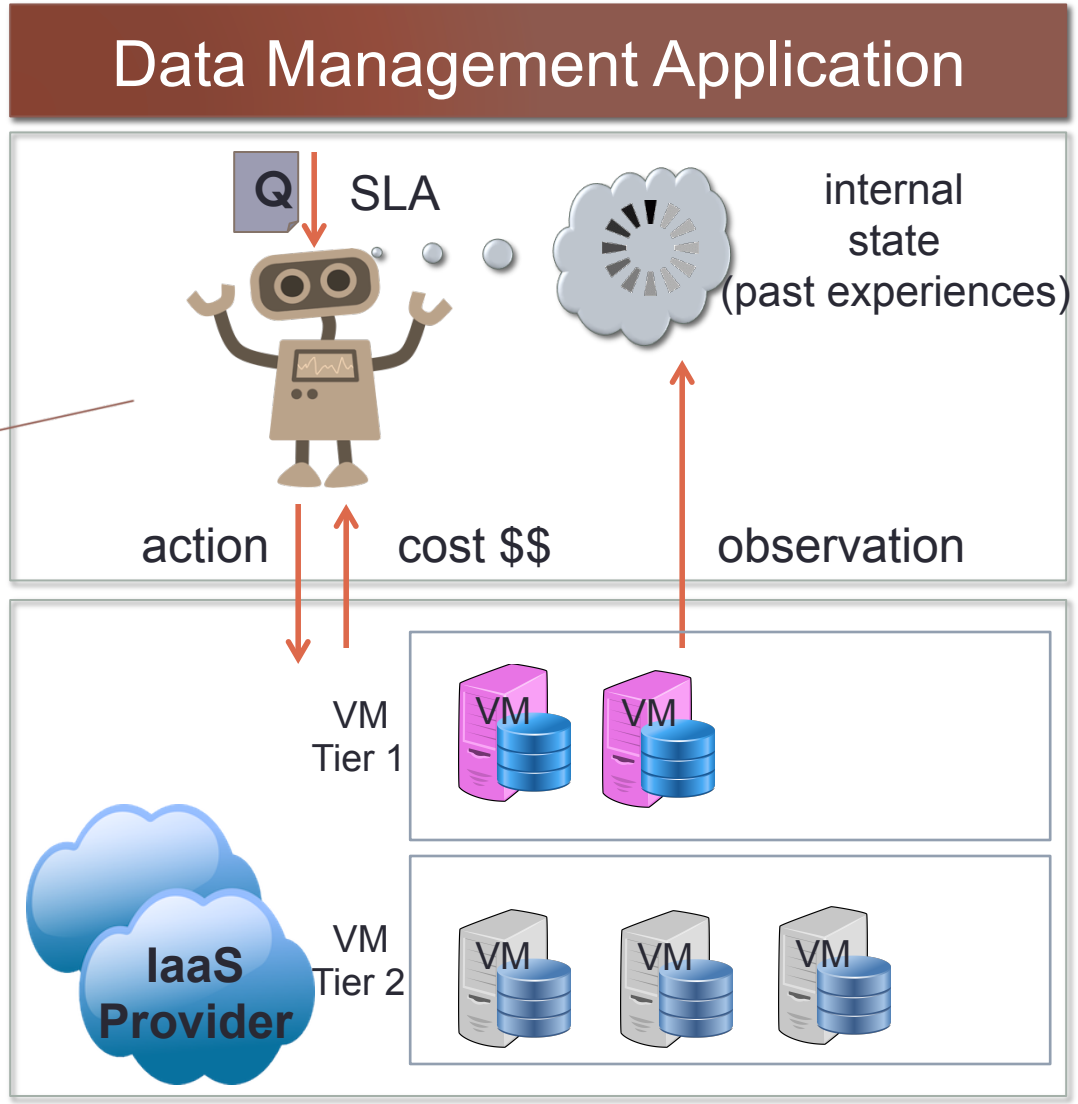
- Accept
- Pass to next /new VM
- Down one VM tier

Reward

- $$ cost: processing & SLA violation penalties

Observation

- context of the decision
- action
- $$ cost



CMABs in Bandit

(Contextual Multi-Armed Bandits)

Action (per VM)

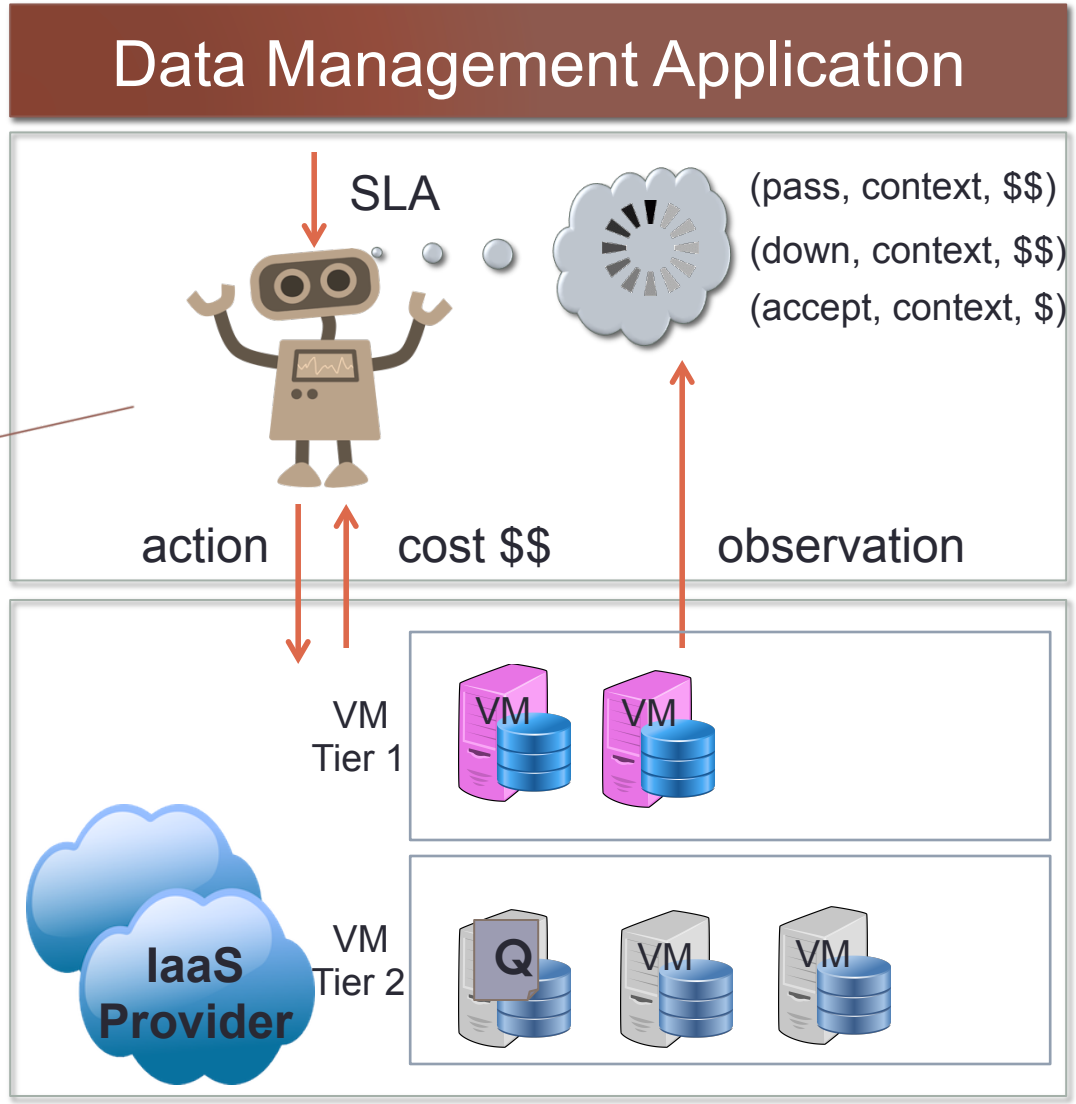
- Accept
- Pass to next /new VM
- Down one VM tier

Reward

- $$ cost: processing & SLA violation penalties

Observation

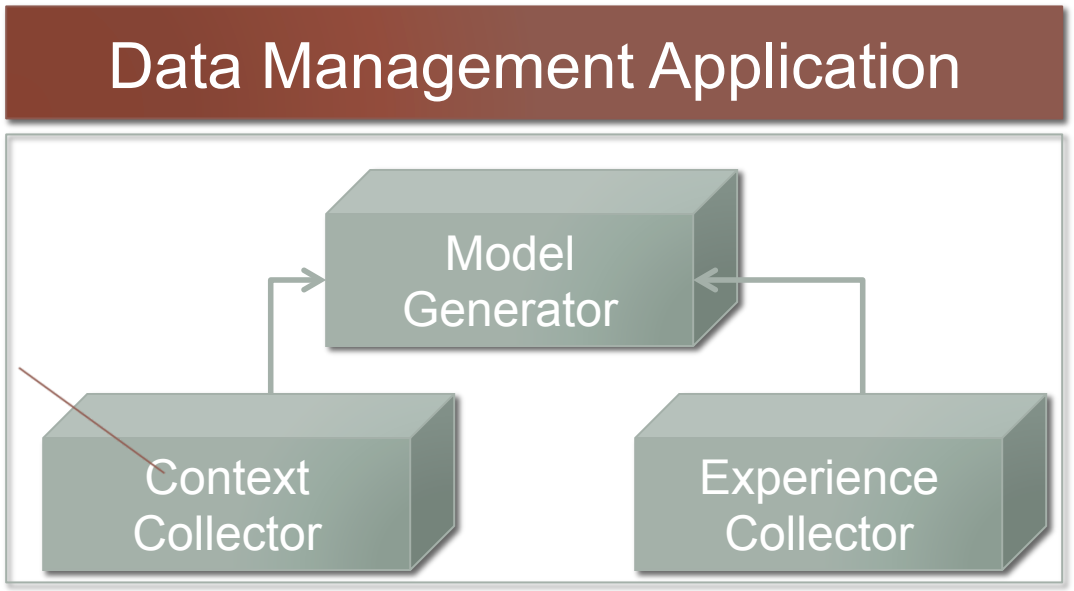
- context of the decision
- action
- $$ cost

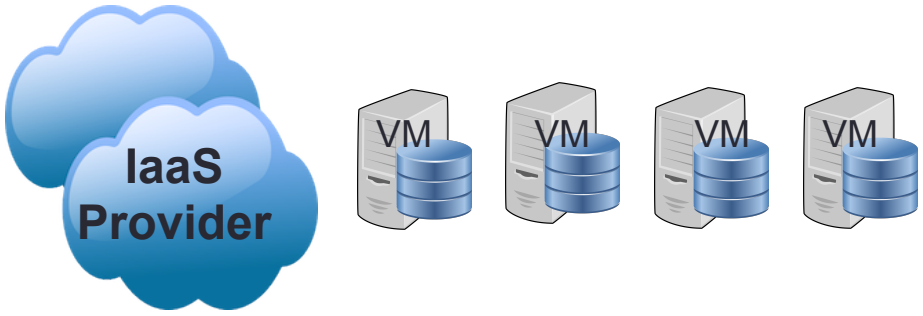


Feature Selection

Context Features (for OLAP)

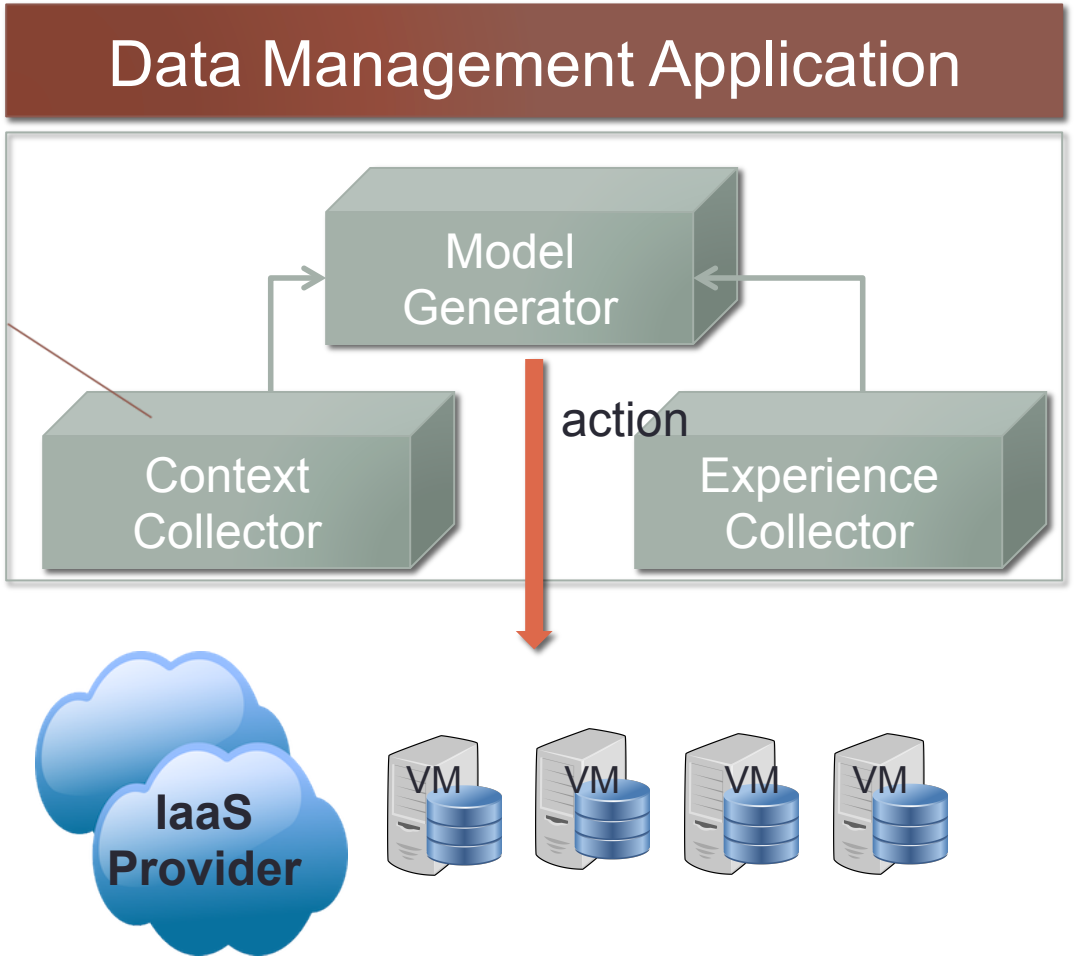
- **VM context**
  - memory, I/O rate
  - #queries in queue
  - network cost (for partitions)
- **Query context**
  - tables used by current query
  - tables used by old query
  - # table scans
  - # joins & # spill joins
  - cache reads in the plan



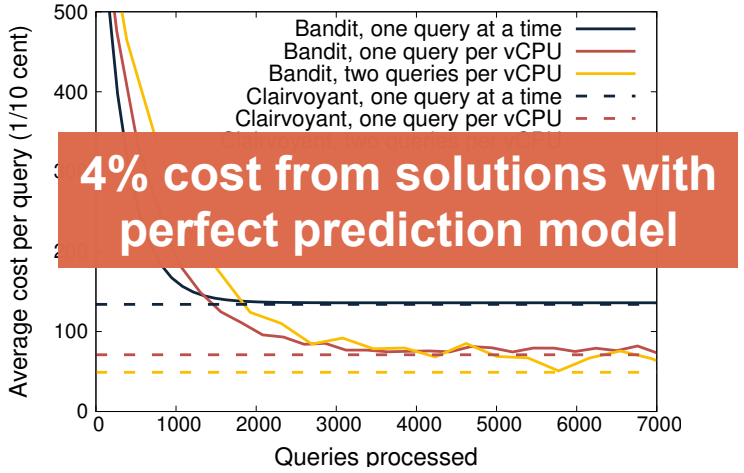


Probabilistic Action Selection

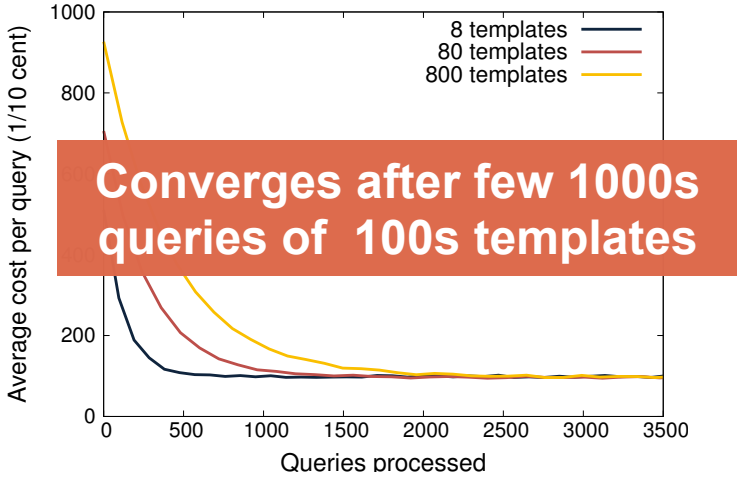
- **Explore** opportunities
  - gather information
- **Exploit** "safe" actions
  - make best decision given current information
- **Balance exploration vs exploitation**:
  - Thompson sampling



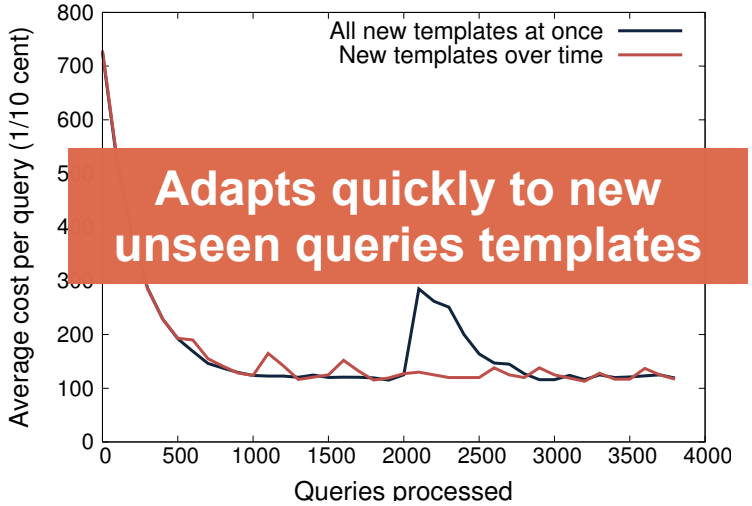
Evaluation



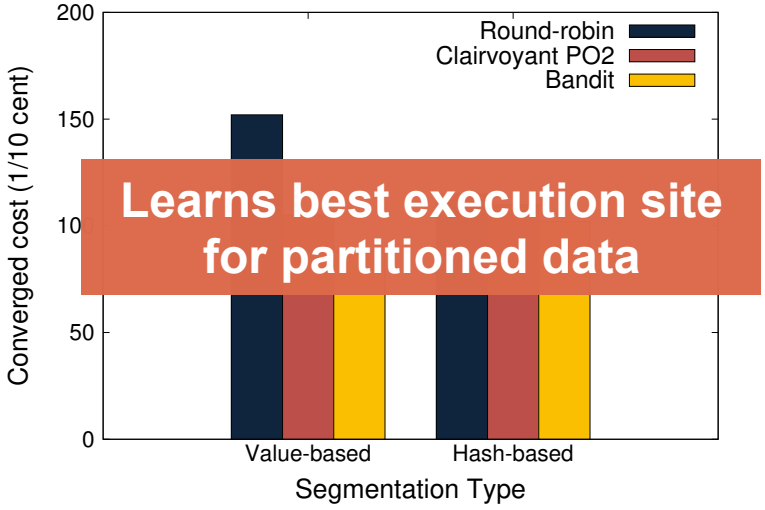
4% cost from solutions with perfect prediction model



Converges after few 1000s queries of 100s templates



Adapts quickly to new unseen queries templates



Learns best execution site for partitioned data

Conclusions

- **Cost vs performance trade-offs are complex**
  - human ability to derive insight is not improving
- **Benefits of ML-drive approach**
  - discover customized solutions
  - automate decision making
  - adapt to dynamic environments
- **Future Steps**
  - alternative learning techniques
  - more advanced tasks: scheduling, data movement
  - learning-based database as a service (DaaS) systems