# Hedge Your Bets: Optimizing Long-term Cloud Costs by Mixing VM Purchasing Options

**Pradeep Ambati, Noman Bashir, David Irwin, Prashant Shenoy**

**University of Massachusetts Amherst**

# Cloud Platforms are Complex

- Cloud platforms are tremendously complex
  - Need to manage physical and virtualized resources in data centers
  - Need to grow or shrink the resources allocated on the fly (scalability)
  - Need to offer multitude of compute services required for supporting applications

- Cloud platforms offer dozens of VMs (compute resources)
  - With different CPU, memory, network, and disk configurations etc
  - Each VM costs different price

# Different Cloud Purchasing Options

- Cloud platforms offers VMs under different purchasing options
  - Time commitment
  - Price
  - Availability guarantees

- Common purchasing options among the popular cloud platforms (AWS EC2, GCP, and Azure)
  - On-demand VM type
  - Reserved VM type
  - Transient VM type

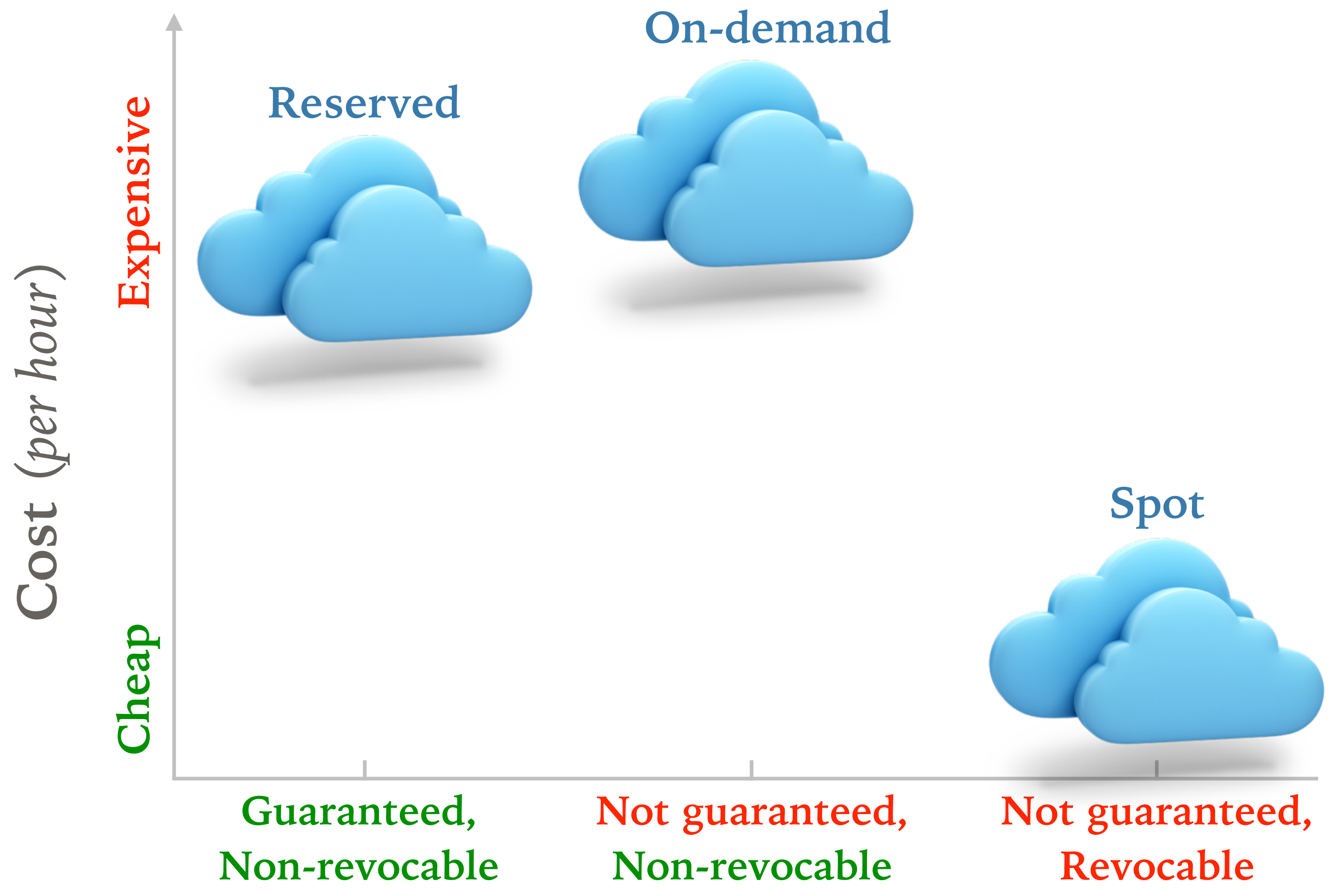
# Different Cloud Purchasing Options

- On-demand VM type
  - *Most popular VM type (usually the default option)*
  - Billed per unit time until terminated
  - No time commitment required

- Reserved VM type
  - *Requires long time commitment (1-year or 3-year)*
  - Billed for entire duration of time committed
  - Provides steep discount compared to on-demand

# Different Cloud Purchasing Options

- Transient VM type
  - *Cloud platforms can revoke them at any time*
  - No time commitment required
  - Usually offered at discount (20% - 90%) compared to on-demand



# Differences among Cloud Providers

- Google cloud platform unique offerings
  - Sustained use discount
  - Customized VMs

- AWS EC2 unique offerings
  - Spot block VMs
  - Scheduled reserved VMs

# Challenge - Optimizing Long-Term Cloud Costs

- Optimizing for long-term cloud costs requires
  - Provisioning right set of VMs
  - Selecting right mix of purchasing options (i.e., time commitment of VMs)

- Provisioning right set of VMs requires knowledge of workload characteristics (resource usage etc)
  - Subject of much research in the past e.g., Cherry-pick NSDI’17

# Problem Statement

- Selecting right mix of purchasing options requires future workload expectations
  - *Requires balancing two key trade-offs*
  - Extracting the greatest savings through longest possible commitments for the VMs
  - Retaining the flexibility to respond to changing workloads

To address the above problem, we design policies to optimize long-term cloud costs by selecting a mix of VM purchasing options based on short- and long-term expectations of workload utilization.

# Contents

- ~~Motivation~~
- ~~Introduction~~
- ~~Background~~
- Design
- Implementation
- Evaluation

# Long Term Cost Optimization Policy

- **Goal**: Select the VMs (and purchasing options) to minimize the long-term cost to execute the workload

- **Assumptions**: Workload consists of batch submissions
  - Batch requests include requested cores and memory size

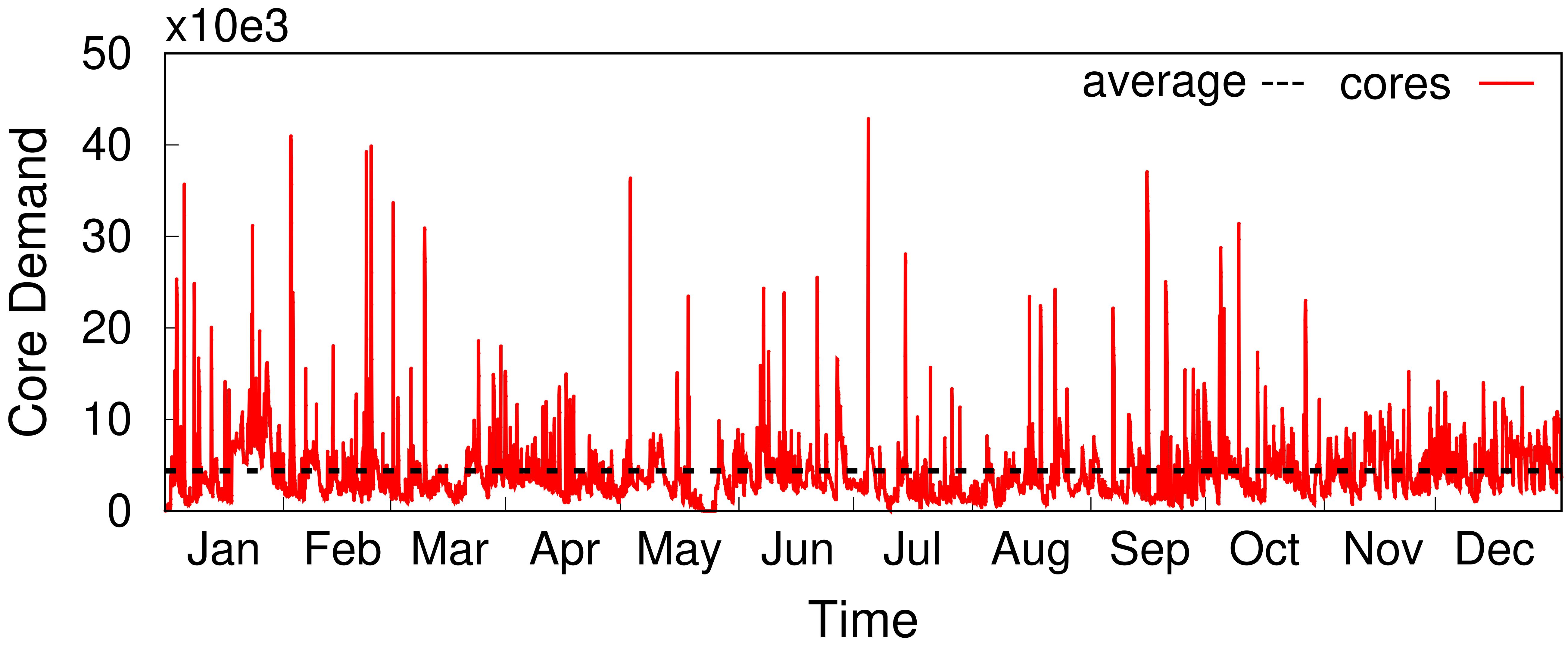
- Design two policies
  - Optimal offline policy
  - Practical online policy

# Optimal Offline Policy - Overview

- Ideal policy that results in absolute minimum value (**OPT**)
  - Cannot optimize/minimize the cost beyond this policy
- Makes **unrealistic** assumptions
  - Perfect knowledge of future workloads
  - Each job can be subdivided into smaller jobs i.e., fractional jobs
- Can be used as a benchmark to compare realistic policies

# Optimal Offline Policy - Basic Approach

- Transform the workload as a time-varying function of resource demand
  - Model workload trace in terms of aggregate resource demand per unit time



# Optimal Offline Policy - Basic Approach

- Transform the workload as a time-varying function of resource demand
  - Model workload trace in terms of aggregate resource demand per unit time

- For each unit of resource pick the cheapest purchasing option
  - Compute the normalized cost under each purchasing option*
  - Select the purchasing option that results in cheapest normalized cost

- Repeat until the demand across all time slots is satisfied

# Normalized Cost - On-demand VMs

- Normalized cost of a VM depends on
  - VM Time commitment
  - VM Revocations
  - Job length (in case of transient VMs)

- **On-demand VMs** (sustained-use): Computing normalized cost is straightforward
  - Simply assign the on-demand cost to each unit of resource demand

# Normalized Cost - Transient VM

- Normalized cost of transient VMs is a function of
  - relative cost per unit time and job length
  - Revocation rate (and fault tolerance)
- Ensure a job assigned to a transient VM completes
  - If a transient VM is revoked, restart the job on on-demand VM
- Derive normalized cost of transient VM using

$$E[C(T)] = (1 - R(T))(p_{\mathtt{transient}} \times T) + R(T)(p_{\mathtt{transient}} \times E_{\mathtt{revoke}}[T] + p_{\mathtt{ondemand}} \times T)$$

# Normalized Cost - Spot Block

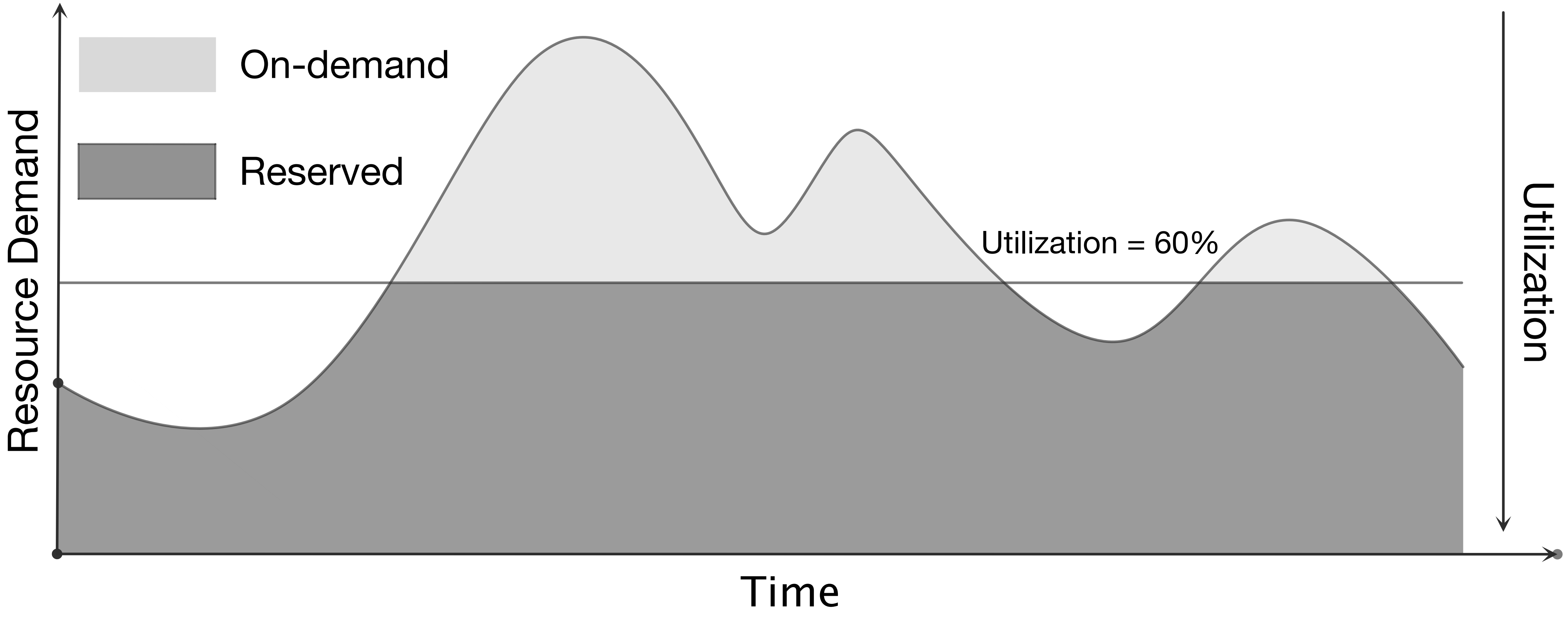
- Spot blocks are available for maximum of 6-hour duration
  - Purchased in 1-, 2- , 3- ,4- , 5-, and 6-hour duration blocks
  - 1-hour block costs 55% of on-demand price
  - 6-hour block costs 70% on-demand price

- In the offline case, if a job can be run on spot block
  - Simply assign the spot block cost to job cost

- Spot block slightly more expensive than the transient option

# Normalized Cost - Reserved VMs

- For reserved VMs, normalize cost by its utilization over the time of the commitment
  - In offline case, we assume reserved VMs as fractional resource



# Normalized Cost - Scheduled Reserved VMs

- Similar to normalized cost of a reserved VM, but…
  - … must consider every possible daily, weekly, and monthly schedule
  - Select multiple schedules that maximize discount
- Problem of finding schedule reduces to weighted-job scheduling
  - In our approach, each possible schedule is akin to a job in weighted-job scheduling
  - The normalized discount of each schedule is akin to its value
  - The output is the cheapest set of schedules

# Selecting Purchasing Option

- Compute normalized costs for different purchasing options…
  - …for each unit of demand at each unit of time
- Determine the cheapest option from
  - Transient, spot block, on-demand, and scheduled reserved
- Compare the cheapest non-reserved to 1-year reserved option
  - Select the best option out of them
- Next, compare best option from above and 3-year reserved
  - In this way, our approach yields lowest cost purchasing option for each unit

# Practical Online Approach

- Online approach is essentially same as offline policy, but …
  - …  utilizes predictions of demand in the place of perfect knowledge
  - … does not assume a fractional supply and demand
  - … map the resources (cores and memory) to specific type of VMs

- Our online approach is a heuristic
  - even given perfect future knowledge of the workload, the problem is NP-hard

- Our approach is biased towards large VMs (reserved)
  - since they are can run larger jobs (and pack multiple jobs)

# Practical Online Approach

- Simply take prior jobs data and apply our offline approach
  - Assume prior jobs data will repeat to estimate the future workload
  - Estimate the reserved capacity to purchase

- After purchasing reserved capacity, as jobs arrive…
  - … Schedule them on available reserved VMs based on resource requirements
  - …. If reserved VMs are not available, dynamically acquire additional non-reserved resources to execute the job

# Implementation

- Implemented our policies using python

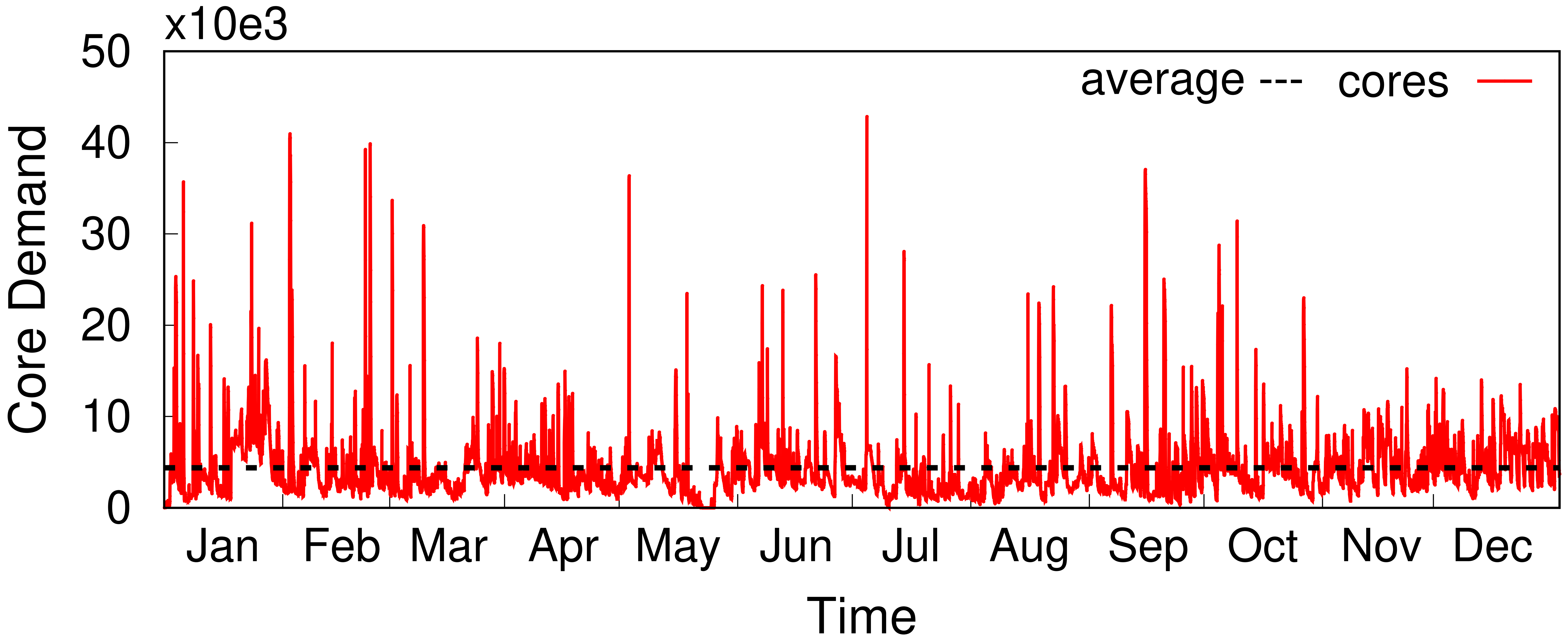
- Dataset consists of a 4-year trace of job submissions from a 14k batch cluster
  - Trace contains ~60 million jobs over the 4-year period (2015-2018)
  - Each job entry includes its submission time, requested cores and memory, and duration

# Evaluation

- Evaluate the cost benefits of using a mix of purchasing options
  - Compare online policy to offline policy
  - Compare both offline and online policy to single purchasing option
  - Compare the cost benefits for the set of unique purchasing options offered by each cloud provider

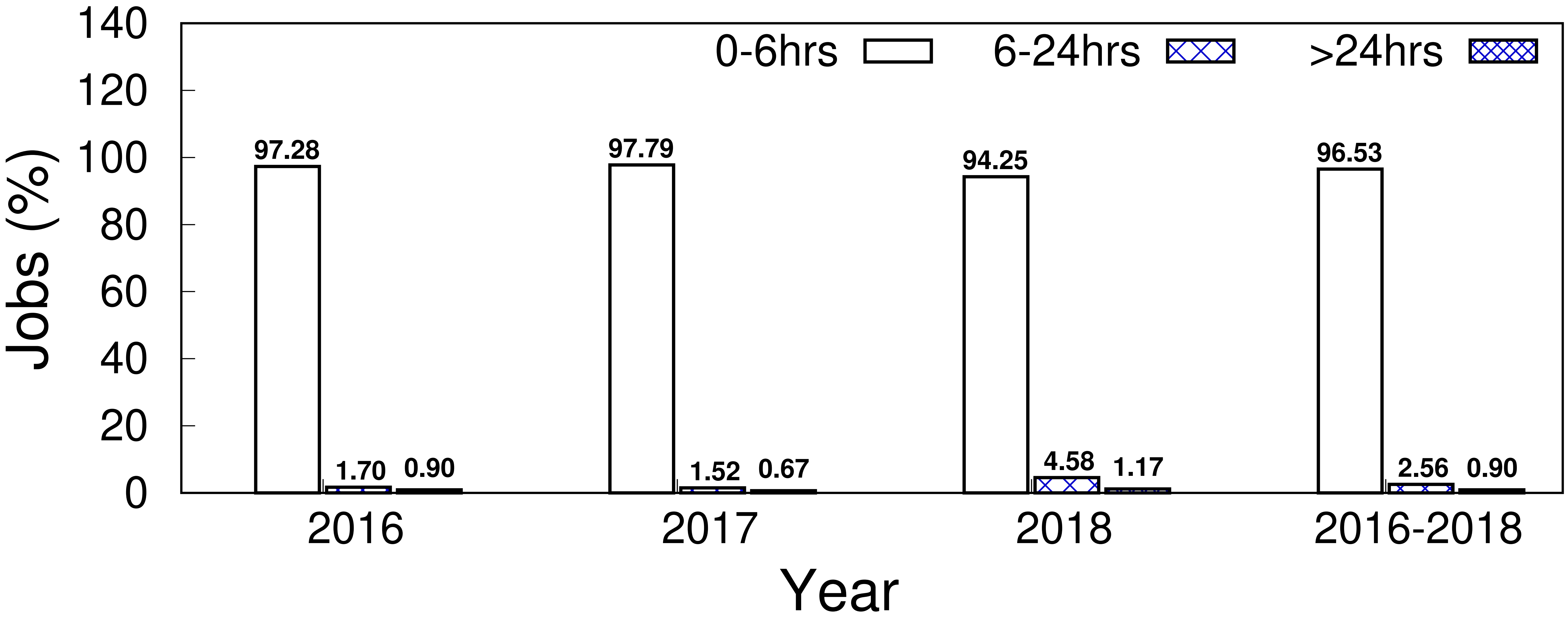
- For online approach evaluation
  - use the first year of jobs for training,
  - evaluate on the next 3 years (2016-2018) of jobs

# Batch Trace Characteristics



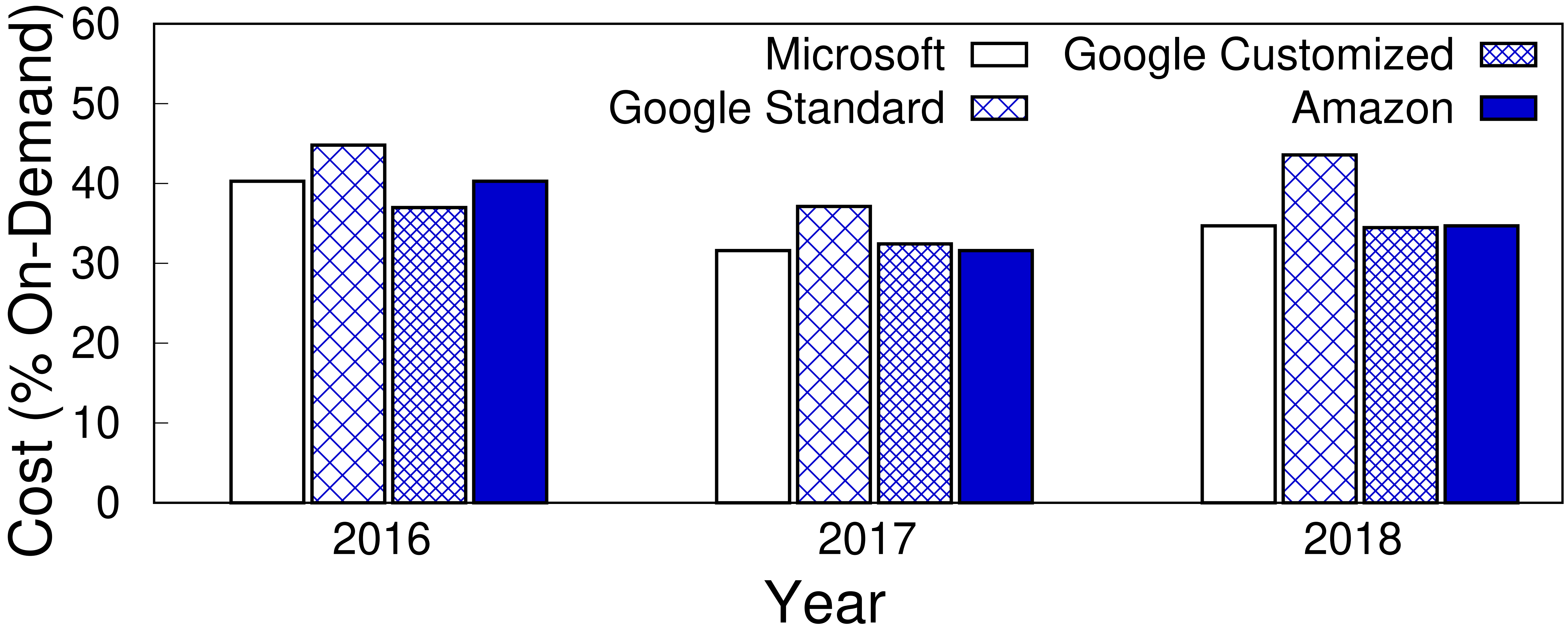

**Trace shows that there are load bursts often during the time period shown**

# Batch Trace Characteristics



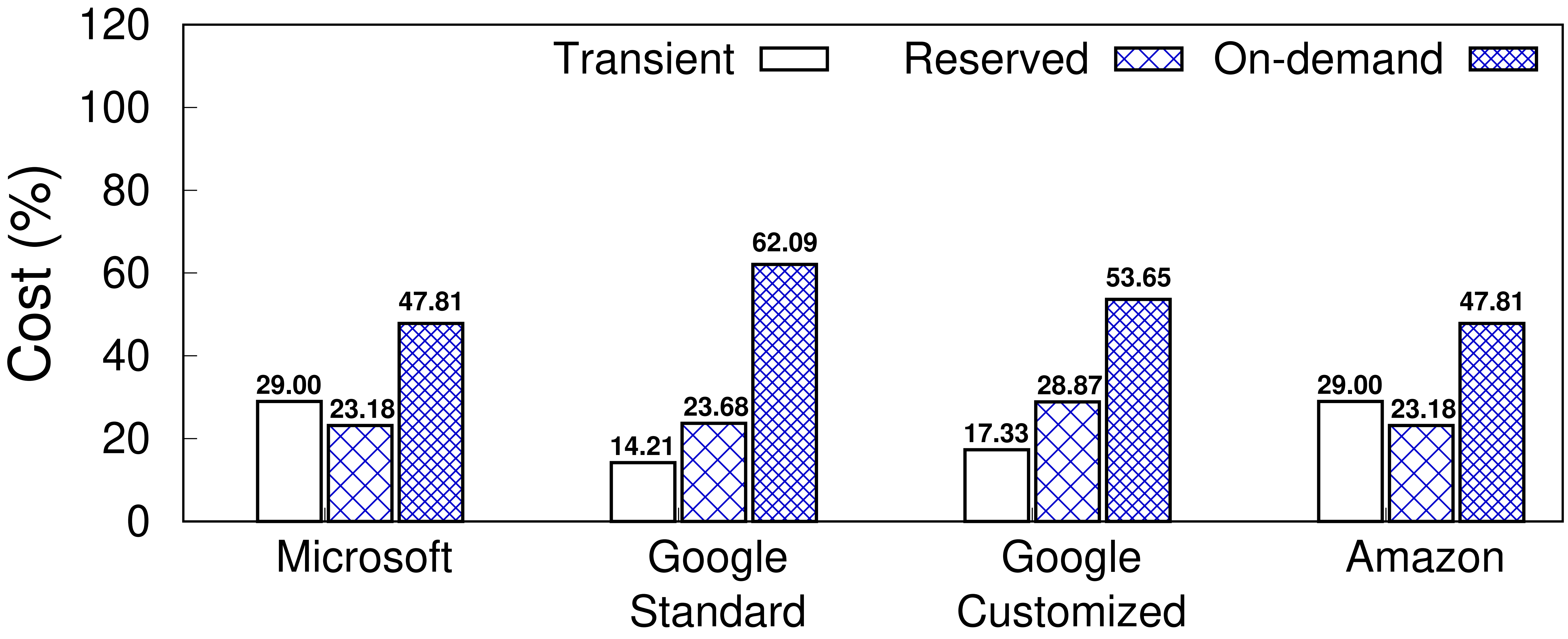

**Our batch trace consists of significant number of short jobs (< 6 hours)**

# Evaluation - Optimistic Offline Policy



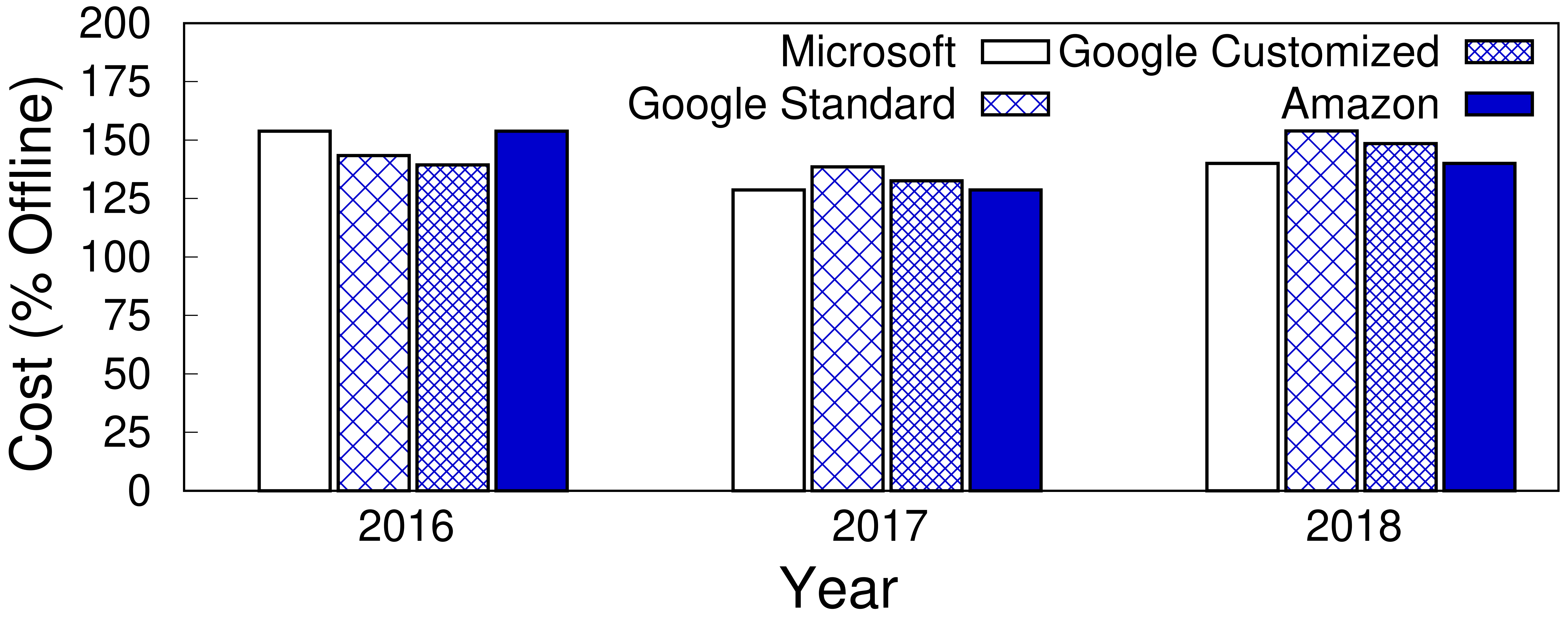

**Our offline policy minimizes cost (relative to on-demand) significantly**

# Evaluation - Optimistic Offline Policy




**On-demand cost dominates the total cost due to the frequent load bursts in the workload**

# Evaluation - Practical Online Approach




**The online approach results in 35-55% greater cost compared to their respective optimistic offline approach**

# Conclusion

- Presented policies to optimize long term cloud costs by selecting a mix of VM purchasing options
  - Accounts for uncertainty in future workload demand
  - Proposed both offline optimal (OPT) and practical online policy

- Evaluated our policies on a batch job trace spanning 4 years
  - Online policy incurs a cost within 41% of an optimistic offline optimal approach
  - Online policy costs 50% less than solely using on-demand VMs, and 79% less than using reserved VMs.