Toward ML-Centric Cloud Platforms (SOSP’17)

By Ricardo Bianchini, Marcus Fontoura, Eli Cortez, Anand Bonde, Alexandre Muzio, Ana-Maria Constantin, Thomas Moscibroda, Gabriel Magalhaes, Girish Bablani, and Mark Russinovich

Presented by: Allen Jue

Cloud Computing

aws

Google Cloud

Cloud Resource Management

- Large-scale datacenters run heterogeneous, co-located workloads
- Resources are oversubscribed (CPU, memory, disk, power)
- Managers must balance:
  - Performance isolation
  - Utilization efficiency
  - Reliability & availability

Traditional Resource Management

- Rule-based
- Heuristics?
- Offline analysis & manual tuning
- Conservative to avoid outages
- Ex: Borg (Verma et al., EuroSys 2015)
  - Google's cluster scheduler

Rise of Machine Learning

Suggested ML Use Cases

- Container scheduler
  - Where to put containers on servers?
- Server defragmenter / migration manager
  - Freed resources… what to do?
- Power capping manager
  - Literally do not use too much power for CPU
- Server health manager
  - When to move things when a server is dying?

Motivation

- ML vs. Traditional Techniques
- Opportunities for ML in Cloud Platforms
- The researchers want to create something that is:
    - **Generalizable, debuggable, available, and maintainable**

Example: Cache Interference

- Consider two co-located containers in the same machine.
- If they share the same cache, certain access patterns may cause throttling.
- **Problems with heuristics!**
- ML can learn throttling patterns to improve performance

Design Dimensions

- Application performance vs. counters
  - What kind of data?
- Predictions vs. Actions
  - Do we want the AI to predict actions or perform them?
- Integration or Separation
  - Merge with manager?
- Immediate or Delayed Feedback
  - Train offline in batches or ASAP?

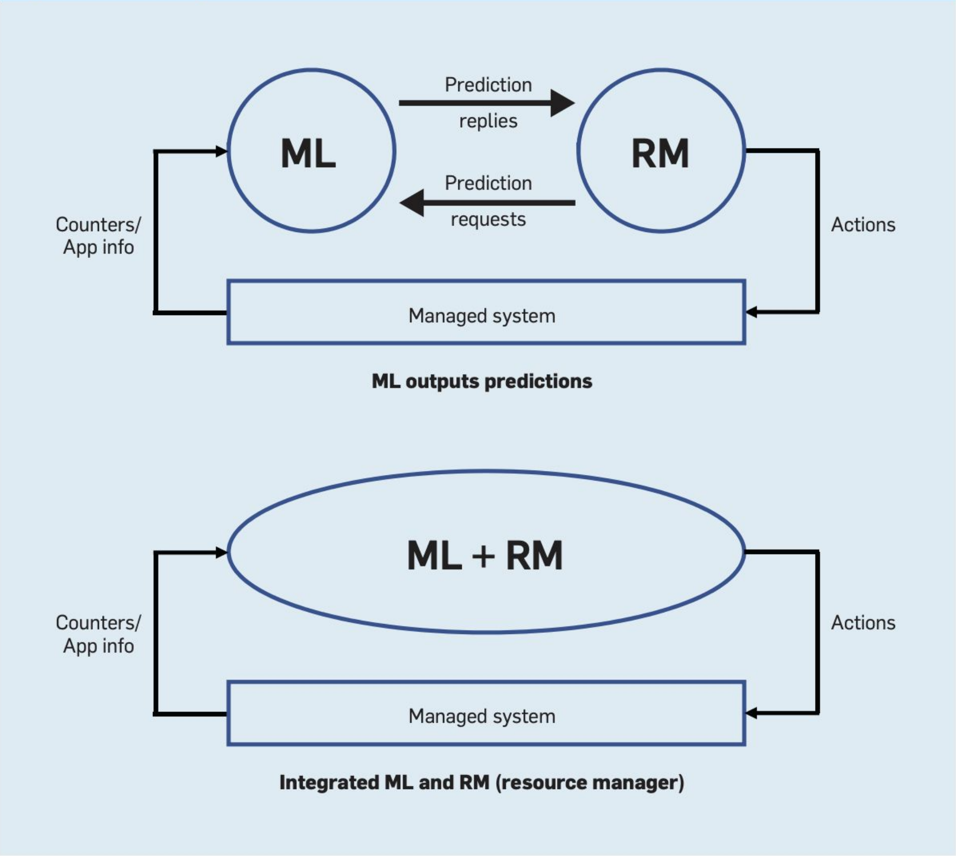
Design Dimensions

- Application performance vs. Counters
  - What kind of data?
- Predictions vs. Actions
  - Do we want the AI to predict actions or perform them?
- Integration or Separation
  - Merge with manager?
- Immediate or Delayed Feedback
  - Train offline in batches or ASAP?

Resource Central (RC)

- RC is not a scheduler
- RC is a central prediction service
- Trained offline
- Use coarse-grained data

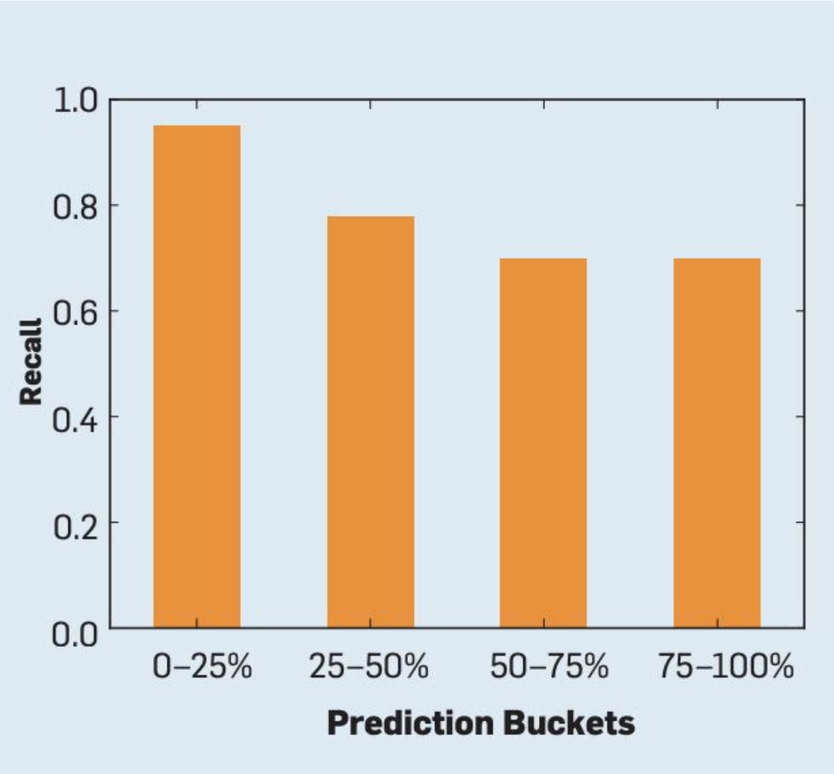
Figure 1. Two designs.



Resource Central (RC)

- Use Gradient Boosting Trees (GBTs)
- Input: Feature vector describing a VM/container at creation time
- Output: Helpful insights to solve fragmentation, capacity planning, migrations
- **It's hard to be exact, so predict buckets** (Ex: 0-24%, 25-49%, …)

Results



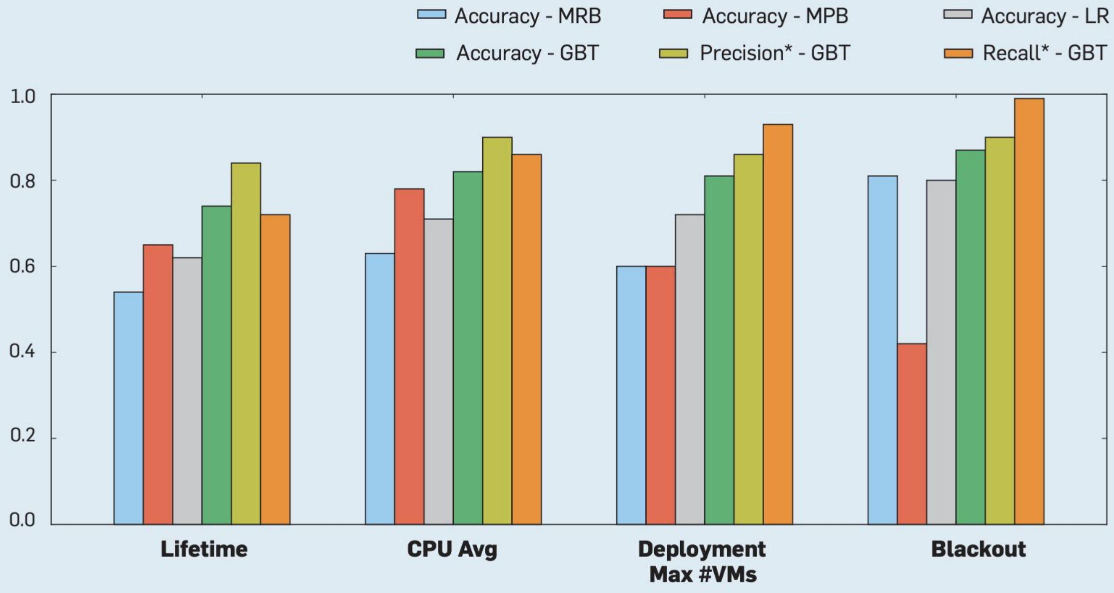
https://github.com/Azure/AzurePublicDataset

- Measure with accuracy, precision, and recall
- VM utilization - Recall between 70-90%
- Accuracy increases when discarding predictions with low confidence.

Results

Figure 4. Accuracy, precision, and recall for all behaviors.

Three leftmost bars of each behavior represent the accuracy for most recent bucket (MRB), most popular bucket (MPB), and logistic regression (LR). Two rightmost bars represent precision and recall with gradient boosting tree (GBT), when predictions with <60% confidence are excluded.



- GBT > Baseline Heuristics
- Accuracy between 84% (lifetime) and 90% (CPU utilization and blackout time)

Discussion

- First time VM deployments were easier to predict (new users deploy in the same way, maybe with a guide)
- Regression seems to be swayed by large, outlier VMs. Buckets > numerical regression.
- During March 2019, 1.5 billion queries were made to RC daily to help make efficient decisions for container management.
- **So are we done?**

Strengths

- They use GBTs as a demonstration - simple and leaves room for optimizations.
- Modular separation of concerns - easier to debug and reproduce.
- Given the lightweight nature of these models + improving hardware, caching can improve performance.

Limitations

- Separation of concerns - limits responsiveness
- Can’t leverage real-time details
- Limited to low-level counters (no instrumentation)
- Lack of democratization
- Opens door to gaming the system?
- What about big models?
- Skirts around RL

Open Research

- Broadly using application-level performance data
- Using action-prescribing ML while being general
- Quick feedback at scale
    - SOL: On-node Learning (Wang et al., ASPLOS '22))
- Other aspects of cloud platforms
    - Azure predictive failure mitigation, Narya (Levy et al., OSDI '20)

Takeaway

- RC demonstrates ML can improve cloud resource management
- Architecture: modular, separable, and extensible
- **NOT THE FINAL ANSWER**