AHAB: Data-Driven Virtual Cluster Hunting

Johannes Zerwas* Patrick Kalmbach* Carlo Fuerst°
Arne Ludwig° Andreas Blenk* Wolfgang Kellerer*
Stefan Schmid^

*Technical University of Munich, Germany
°Technical University of Berlin, Germany
^University of Vienna, Austria

IFIP Networking 2018, Zurich, Switzerland
• Increased use data-intensive applications in shared data centers
• Many provider-tenant interfaces neglect network as a resource
• Problems:
  – Unpredictable application performance
  – Limited applicability of cloud
  – Inefficiencies in production data centers

• Solution: Network-aware abstraction - Virtual Cluster
  (ACM SIGCOMM 2011)
Background: Virtual Cluster Abstraction

Physical Cluster
- Compute Units (CUs)
- Bandwidth Units (BUs)
- Tree-like topology (abstracted from Fat-Tree)

Virtual Cluster (VC)
- Number of VMs (N)
- Size of VMs (S)
- Bandwidth (B)
- Lifetime given resource fulfillment
Background: Virtual Cluster Abstraction

Physical Cluster
- Compute Units (CUs)
- Bandwidth Units (BUs)
- Tree-like topology (abstracted from Fat-Tree)

Virtual Cluster (VC)
- Number of VMs (N)
- Size of VMs (S)
- Bandwidth (B)
- Lifetime given resource fulfillment

Footprint \( F = 6 \)

Utilization \( U = \frac{9}{32} \)
Problem: Resource Fragmentation

Existing allocation algorithms focus on single request:
- Oktopus (ACM SIGCOMM 2011)
- Kraken (IEEE/ACM TON 2018)
TETRIS: Sacrifice Footprint for Fragmentation

Choose hosts with max. ratio of residual resources

\[ t = \frac{4 - 2}{4 - 1} \]
Choose hosts with max. ratio of residual resources
Algorithm Evaluation

- **Baseline**: OKTOPUS (ACM SIGCOMM 2011), KRAKEN (IEEE/ACM TON 2018)

- **Physical Cluster**: Fat-Tree with $k=12$, 8CUs and 8BUs

- **Performance metrics**: CU Utilization, avg. VC Footprint

- **Virtual Cluster Requests**:
  - 1000 / run with varying arrival rates
  - Num. VMs, size VMs, BW similar to traces from Google & Microsoft
TETRIS Evaluation

- +5% utilization
- +10% footprint
TETRIS Evaluation

Add Admission Control
AHAB: The Case for Data-Driven Admission Control

Leverage Knowledge
Monte Carlo Tree Search
Data-Driven Decision

Johannes Zerwas (TUM)
AHAB: The Case for Data-Driven Admission Control
AHAB: The Case for Data-Driven Admission Control

Utilization

accept
reject

1
1

1
1

1
1

... = 12

1/2
1/4
1/4
0/2

1/2
1/2
1/2

1/2
1/2
1/2

1/2
1/2
1/2

1/2
1/2
1/2

Utilization

Johannes Zerwas (TUM)
AHAB: The Case for Data-Driven Admission Control

Works with every VC embedding algorithm (Oktopus, Kraken, Tetris)
AHAB improves utilization

**Graphs showing:***

- **CU utilization**: With AHAB, there is a +10% utilization increase compared to baseline.
- **Avg. F(VC)**: AHAB reduces footprint by -25%.

**Legend:**
- Oktopus
- Tetris
- AHAB(Kraken)
- AHAB(Oktopus)
- AHAB(Tetris)
- Kraken
Why is AHAB better?

**Kraken**

- **Bandwidth (BU)**
  - 1
  - 2
  - 4
  - 8

- **Num. VMs**
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8

- **Size VMs (CU)**
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8

- **Acceptance Ratio**
  - 0.0
  - 0.2
  - 0.4
  - 0.6
  - 0.8
  - 1.0

**AHAB(Kraken)**

- **Bandwidth (BU)**
  - 1
  - 2
  - 4
  - 8

- **Num. VMs**
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8

- **Size VMs (CU)**
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8

- **Acceptance Ratio**
  - 0.0
  - 0.2
  - 0.4
  - 0.6
  - 0.8
  - 1.0

**Small VMs**

- Large BW

**Large VMs**

- Small BW
Why is AHAB better?

AHAB accepts more valuable requests.
Optimization Opportunities

Trade-Off: Utilization - Computations

Use ML for speed-up
Summary

- **TETRIS** sacrifices footprint increase utilization
- **AHAB** employs a data-driven approach for Admission Control
- **AHAB** evaluates the impact of a single request on future requests
- **AHAB’s** approach applies also to other use-cases
- Future Work: Use ML to predict **AHAB’s** decisions
Thank you!
Questions?