TUNING GUIDE

Microsoft Azure Virtual Machines (VMs) Powered by AMD EPYC™ Processors

Publication	63852
Revision	1.0
Issue Date	November, 2024

Date	Version	Changes
Nov, 2024	1.0	Initial public release

Audience

This document is intended for a technical audience such IT professionals, cloud administrators, and developers who have:

- Admin access to the Microsoft Azure VMs.
- Knowledge and experience selecting and configuring Microsoft Azure VMs for various workloads.
- Admin OS access.

Author

Zhang Zhang

Table of Contents

Chapter 1 Introduction

This guide describes best practices for IT professionals, cloud architects, and developers to deploy and optimize Microsoft Azure virtual machines (VMs) powered by AMD EPYC processors. The guidelines contained herein will help drive informed decisions about VM selection and configuration.

Microsoft offers a wide range of VMs powered by AMD EPYC processors that deliver excellent price-performance across general-purpose, compute-optimized, memory-optimized, storage-optimized, burst computing, confidential computing, and high-performance computing (HPC) use cases. These VMs include:

AMD EPYC Generation	VM Types
1st (AMD EPYC 7001 Series)	• LSv2
2nd (AMD EPYC 7002 Series)	• HBv2 • Dav4 / Dasv4 • Eav4 / Easv4
3rd (AMD EPYC 7003 Series) Please see *Overview of AMD EPYC™ 7003 Series Processors Microarchitecture* (available from the AMD Documentation Hub) to learn more about 3rd Gen AMD EPYC processors.	• Lasv3 • HBv3 • Dasv5 / Dadsv5 • DCasv5 / Eadsv5 • Easv5 / Eadsv5 • Basv2
4th (AMD EPYC 9004 Series) Please see *AMD EPYC™ 9004 Series Architecture Overview* (available from the AMD Documentation Hub) to learn more about 4th Gen AMD EPYC processors.	• HBv4 • HX • Dasv6 / Dadsv6 • Dalsv6 / Daldsv6 • Easv6 / Eadsv6 • Fasv6 / Falsv6 / Famsv6

Table 1-1: Microsoft Azure VM types by processor generation

Note: Future editions of this tuning guide will include Microsoft VM families with GPU acceleration.

1.1 Microsoft Azure VM Families

Table 1-2 compares the Microsoft Azure VM families listed in Table 1-1, above, and helps inform VM family selection by highlighting the specialized nature of each VM type.

Feature	D-Series (General Purpose)	E-Series (Memory Optimized)	F-Series (Compute Optimized)	L-Series (Storage Optimized)	HB-Series (HPC VMs)
Latest Generation	V6	V6	V6	V6	V4
Processor Generation	4th Gen AMD EPYC				
Max vCPUs	96		64	96	176
SMT	ON		OFF	ON	OFF
Max Memory	384 GiB	672 GiB	512 GiB	384 GiB	768 GiB
Memory::vCPU Ratio	2::1 or 4::1	8::1 (7::1 for the largest shape)	2::1, 4::1, or 8::1	8::1	768 GiB regardless of shape
Max Network BW	40 Gbps		36 Gbps	32 Gbps	400 Gbps (RDMA)
Max Remote Disk BW	4.32 Gbps		2.88 Gbps	1.4 Gbps	12 Gbps
Local Storage	Up to 6 local NVMe (optional)		N/A	Up to 19.2 TB local NVMe	• 2 x 1.8 TB NVMe • 1 x 480 GB temp
Remote Storage	Up to 32 disks				
Use Cases	• Web servers • Small-medium databases • Dev environments	• Traditional databases • In-memory databases • Distributed caches • Big Data analytics	• Batch processing • Ad serving • Media and Entertainment • Financial Services • Gaming	• NoSQL database • Real-time analytics • Large-scale log analysis	• HPC workloads • AI/ML
AVX-512 Support	Yes				
Comments	D-series and E-series VMs are equivalent in many aspects, except that E-series VMs feature a higher memory-to-core ratio.		SMT=OFF maps each vCPU to a full physical core for high, uncontested performance. These full-core VMs suit workloads demanding high CPU perf.	These VMs are optimized to use the local disk on the node attached directly to the VM rather than using durable data disks for high IOPS and throughput.	400 Gbps NDR InfiniBand enables supercomputer-scale MPI workloads. High memory plus local NVMe and temp storage are great for AI/ML inference.

Table 1-2: Azure VM family comparison

1.1.1 General Purpose (D-Series)

Table 1-3 highlights some of the differences between generations of general purpose Microsoft Azure D-series VMs.

Feature	4th Gen AMD EPYC	3rd Gen AMD EPYC	2nd Gen AMD EPYC
Latest Generation	Dasv6*	Dasv5*	Dasv4*
Max Frequency	3.7 GHz	3.5 GHz	3.35 GHz
Max vCPUs	96		
Max Memory	384 GiB		
Max Network BW	40 Gbps		
Max Disks	32		
Max Disk IOPS	175,000	80,000	
Supported Processor Capabilities	• AVX3-512 • VNNI • bloat16	AVX2	

Table 1-3: Generational comparison of Microsoft Azure D-series VMs

D-series Microsoft Azure VMs include additional variants that provide finer granularity when choosing VMs for applications, including:

- The Dav4 variant is comparable to the Dasv4 variant but does not support premium storage, which may optimize costs for applications that do not need high performance storage.
- The Basv2 variant is comparable to the Dasv5 variant except that it supports burstable performance, making it cost effective for applications that require sporadic CPU performance bursts during execution.
- The Dadsv5 and Dadsv6 variants are comparable to the Dasv5 and Dasv6 variants, respectively, and come with either one (Dadsv5) or up to six (Dadsv6) locally attached disks. They are suitable for applications that require storage with high throughput and low latency.
- The Dalsv6 variant is comparable to the Dasv6 variant except that it offers a lower ratio of 2 GiB per vCPU instead of 4 GiB per vCPU.
- The Daldsv6 variant combines the features of both the Dadsv6 and Dalsv6 variants.

1.1.2 Memory Optimized (E-Series)

Table 1-4 highlights some of the differences between generations of memory optimized Microsoft Azure E-series VMs.

Feature	4th Gen AMD EPYC	3rd Gen AMD EPYC	2nd Gen AMD EPYC
Latest Generation	Easv6*	Easv5*	Easv4*
Max Frequency	3.7 GHz	3.5 GHz	3.35 GHz
Max vCPUs	96		
Max Memory	672 GiB		
Max Network BW	40 Gbps		
Max Disks	32		
Max Disk IOPS	175,000	80,000	
Supported Processor Capabilities	• AVX3-512 • VNNI • bloat16	AVX2	

Table 1-4: Generational comparison of Microsoft Azure E-series VMs

E-series Microsoft Azure VMs include additional variants that provide finer granularity when choosing VMs for applications, including:

- The Eav4 variant is comparable to the Easv4 variant but does not support premium storage, which may optimize costs for applications that do not need high performance storage.
- The Eadsv5 and Eadsv6 variants are comparable to the Easv5 and Easv6 variants, respectively, and come either one Eadsv5) or up to six (Eadsv6) locally attached disks. They are suitable for applications that require storage with high throughput and low latency.

1.1.3 Compute Optimized (F-Series)

Table 1-5 highlights some of the differences between generations of compute optimized Microsoft Azure F-series VMs.

Feature	4th Gen AMD EPYC
Latest Generation	Fasv6*
Max Frequency	3.7 GHz
Max vCPUs	64
SMT	OFF
Max Memory	256 GiB
Max Network BW	36 Gbps
Max Disks	32
Max Disk IOPS	115,200
Supported Processor Capabilities	• AVX3-512 • VNNI • bloat16

Table 1-5: Generational comparison of Microsoft Azure F-series VMs

F-series Microsoft Azure VMs include additional variants that provide finer granularity when choosing VMs for applications, including:

- The Falsv6 variant is comparable to the Fasv6 variant except that it offers a lower ratio of 2 GiB per vCPU instead of 4 GiB per vCPU.
- The Famsv6 series is comparable to the Fasv6 variant except that it offers a higher ratio of 8 GiB per vCPU instead of 4 GiB per vCPU.

1.1.4 Storage Optimized (L-Series)

Table 1-6 highlights some of the differences between generations of storage optimized Microsoft Azure L-series VMs.

Feature	3rd Gen AMD EPYC	1st Gen AMD EPYC
Latest Generation	Lasv3*	Lsv2
Max Frequency	3.5 GHz	3.0 GHz
Max vCPUs	80	80
Max Memory	640 GiB	
Max Network BW	32 Gbps	16 Gbps
Max Disks	Up to 10 x 1.92 TB local NVMe	
Max Disk IOPS	80,000	
Supported Processor Capabilities	AVX2	

Table 1-6: Generational comparison of Microsoft Azure L-series VMs

1.1.5 High Performance Computing (HB-Series)

Table 1-6 highlights some of the differences between generations of compute optimized Microsoft Azure HPC-series VMs. This VM family is designed for compute-intensive workloads. High-performance AMD EPYC processors and fast memory on HB-series VMs offer exceptional CPU and memory bandwidth.

Feature	4th Gen AMD EPYC	3rd Gen AMD EPYC	2nd Gen AMD EPYC
Latest Generation	HBv4*	HBv3*	HBv2*
Max Frequency	3.7 GHz	3.5 GHz	3.3 GHz
Max vCPUs	176	120	120
Max L3 Cache	2.3 GiB	1.536 GiB	480 MB
Max Memory	768 GiB regardless of VM size	448 GiB regardless of VM size	480 GiB (4 GiB/vCPU)
Max Memory BW	780 Gbps (amplified to 5.7 Tbps)	350 Gbps (amplified to 630 Gbps)	350 Gbps
Max Network BW	400 Gbps HDR Infiniband	200 Gbps HDR InfiniBand	
Local Disks	2 x 1.8 TB NVMe	2 x 960 GB NVMe	1 x 960 GB NVMe
Max Remote Disks	32		8
Max Disk IOPS	175,000	80,000	
Supported Processor Capabilities	• AVX3-512 • VNNI • bloat16	AVX2	

Table 1-7: Generational comparison of Microsoft Azure HB-series VMs

The HX family of VMs are a variant of the HBv4 that increase the amount of memory from 768 GiB to 1.4 TiB of memory regardless of vCPU count.

Chapter 2

VM Selection, Optimization, and Sizing

Selecting the right AMD EPYC Microsoft Azure VM type and size is crucial for optimal performance, cost-efficiency, and resource utilization. The selection process involves carefully considering your workload characteristics, performance requirements, and budget constraints to ensure that your applications run efficiently while controlling costs.

2.1 Workload Considerations

Understanding the nature of your workload is the first step in selecting the ideal AMD EPYC VM. Different types of applications have varying compute, memory, storage and network demands. Analyzing your workload profile helps you match your workload to the most suitable Microsoft Azure VM type and size for optimal performance and cost-effectiveness. This section explores various workloads and their corresponding VM recommendations. You can also use the EPYC VM Advisor tool.

2.1.1 Compute-Intensive

Compute-intensive workloads involve complex calculations, simulations, or algorithms that place heavy demands on CPU processing power. They typically have high CPU utilization and benefit from processors with high clock speeds, multiple cores, and advanced instruction set capabilities. Some examples include:

- Scientific computing (e.g., weather modeling, physics simulations)
- Financial modeling and risk analysis
- Video encoding, transcoding, and/or rendering
- Cryptography and encryption
- Machine learning training and inference

Suggested Microsoft Azure VM types:

- HBv4 (4th Gen AMD EPYC)
- Fasv6 (4th Gen AMD EPYC)
- HBv3 (3rd Gen AMD EPYC)

2.1.2 Big Data and Analytics

Big data and analytics workloads involve processing and manipulating large datasets that often require significant memory resources. These applications typically have high memory usage and CPU utilization with frequent data ingestion and transformation operations. Some examples include:

- Data processing and analysis
- Real-time data processing
- Stream processing
- In-memory databases and caching systems
- Business intelligence tools

Suggested Microsoft Azure VM types:

- Easv6 (4th Gen AMD EPYC); for memory-intensive workloads.
- Dasv6 (4th Gen AMD EPYC); for balanced compute and memory needs.

2.1.3 Databases

Database workloads involve frequent disk read/write operations and can be both memory and I/O intensive. These applications constantly read and write data to disk for queries, transactions, and logging, often requiring a balance of compute, memory, and storage resources. Some examples include:

- Relational databases (MySQL, PostgreSQL, Oracle)
- NoSQL databases (MongoDB, Cassandra)
- In-memory databases (Redis, Memcached)

Suggested Microsoft Azure VM types:

- Lasv3 (3rd Gen AMD EPYC); for I/I-intensive and latency-critical workloads.
- Dasv6 (4th Gen AMD EPYC); for balanced database workloads.
- Easv6 (4th Gen AMD EPYC); for balanced database workloads.

2.1.4 Web and Application Servers

Web and application server workloads typically require a balance of compute, memory, and network resources. These applications handle multiple concurrent connections and may experience varying loads throughout the day. Some examples include:

- Web servers (Apache, NGINX)
- Ecommerce platforms
- Cloud-native applications (containerized microservices)

Suggested Microsoft Azure VM types:

- Dasv6 (4th Gen AMD EPYC); for general purpose workloads.

2.1.5 AI/ML Workloads

AI/ML workloads can be both compute and memory-intensive, depending on the specific task. These applications often involve processing large datasets and performing complex mathematical operations. Some examples include:

- Machine learning model training
- Deep learning and neural networks
- Natural Language Processing (NLP)
- Computer vision and image recognition
- Recommendation systems

Suggested Microsoft Azure VM types:

- HBv4 (4th Gen AMD EPYC); ideal because of high core count, high memory bandwidth, and large memory.

2.1.6 High Performance Computing (HPC)

HPC workloads require massive parallel processing capabilities and low-latency networking. These applications typically involve solving complex computational problems that demand high levels of processing power and memory bandwidth. Some examples include:

- Computational fluid dynamics (CFD)
- Molecular dynamics simulations
- Genomics and bioinformatics
- Financial risk modeling
- Seismic analysis in oil and gas exploration

Suggested Microsoft Azure VM types:

- HBv4 (4th Gen AMD EPYC)

2.2 Performance Considerations

Optimizing the performance of Microsoft Azure VMs powered by AMD EPYC processors requires a deep understanding of the processor architecture and carefully tuning various system components. This section explores key performance considerations that can help you maximize the efficiency and throughput of Microsoft Azure VMs powered by AMD EPYC processors, including strategies for optimizing the CPU, memory, I/O, storage, network, and operating system. Implementing the best practices described in this section helps your applications fully leverage the advanced features of AMD EPYC processors such as their high core counts, large L3 cache sizes, and ample memory bandwidth. Performance optimization is an iterative process that requires ongoing benchmarking to determine the best configuration for your application.

2.2.1 CPU Optimizations

Optimizing CPU performance is crucial for compute-intensive workloads. Some key strategies and techniques for maximizing CPU performance include:

1. Identify CPU-bound workloads:
    - Use `htop` to monitor CPU usage. Consistently high utilization (near 100%) indicates CPU-bound processes.
    - Check load averages in `htop`. If they significantly exceed the number of CPU cores/threads, then the system is struggling with CPU demand.

2. Download and install the AMD Optimizing CPU Libraries (AOCL):
   - Set the `AOCL_ROOT` environment variable to point to the AOCL installation directory.
   - Include relevant header files and link against AOCL libraries during compilation:
     ```
     $ gcc -I$AOCL_ROOT/include -L$AOCL_ROOT/lib -lamdlibm -lm your_program.c -o
     your_program
     ```
   - Use specific flags for different optimizations:
     > **Vector math:** `-lamdlibm -fveclib=AMDLIBM -lm`

     > **Faster math:** `-lamdlibm -fsclrlib=AMDLIBM -lamdlibmfast -lm`
3. Maximize L3 cache usage:
   - Use VMs with 16 or more vCPUs for larger exclusive L3 cache access.
   - Group or pin threads that share data to the same L3 cache domain using CPU affinity techniques:
     ```
     $ taskset -c 0-3 your_application
     ```
   - Use CPU pinning to avoid OS process migration away from hot L3 cache data.
4. Optimize Docker container performance:
   - Identify CPU topology using `lscpu` or `lstopo`.
   - Set CPU affinity for Docker in `/etc/docker/daemon.json`:
     ```
     {
       "cpu-rt-runtime": 950000,
       "cpu-rt-period": 1000000,
       "default-cpu-rt-runtime": 950000
     }
     ```
   - Pin containers to specific CPUs:
     ```
     $ docker run --cpuset-cpus="1,3" my-container
     ```
5. Use the Performance CPU governor:
   ```
   $ sudo cpupower frequency-set -g performance
   ```
6. Enable profiling for performance analysis:
   ```
   $ export AOCL_PROFILE=1
   ```
7. Run your application and analyze the generated `aocl_profile_report.txt`.
8. Leverage advanced instruction sets by using compiler flags to enable AVX2 and AVX-512 instructions:
   ```
   $ gcc -mavx2 -mavx512f your_program.c -o your_program
   ```
9. Optimize for specific AMD EPYC generations:
   - Use `-march=znver4` for Microsoft Azure VMs powered by 4th Gen EPYC processors.
   - Use `-march=znver3` for Microsoft Azure VMs powered by 3rd Gen EPYC processors.
   - Use `-march=znver2` for Microsoft Azure VMs powered by 2nd Gen AMD EPYC processors.

2.2.2 Memory Optimizations

Memory-intensive workloads involve processing and manipulating large datasets that must be loaded into memory for efficient access. These workloads thus require systems with large amounts of RAM to avoid excessive paging or swapping to disk, which can significantly degrade performance. Examples of memory-intensive workloads include:

- In-memory databases and caching systems
- Big data analytics and data mining
- Machine learning inference (e.g., recommendation systems)
- Real-time data processing and stream processing
- High-performance computing (HPC) applications

Some key memory optimization strategies include:

1. Maximize memory bandwidth.
   - Resize the VM to utilize all memory channels or full socket:
     - Each AMD EPYC processor Core Chiplet Die (CCD) has roughly 80 GB/s peak usable bandwidth to the I/O die and an exclusive 32MB L3 cache.
     - The peak read bandwidth is approximately twice the write bandwidth, with a total peak usable bandwidth of around 10 GB/s per core.
     - Use larger VM sizes (e.g., E96adsv6 with 96 cores) to fully utilize the socket capacity.
   - Spread applications across multiple CCDs by distributing memory allocations across all NUMA nodes to maximize bandwidth utilization:
     ```
     $ numactl --interleave=all your_application
     ```
   - Leverage Microsoft Azure VMs powered by latest available generation AMD EPYC generations. For example, 4th Gen AMD EPYC processors offer up to 12 DDR5-4800 memory channels per socket for a peak raw memory bandwidth of up to 460 GB/s per socket.
2. Enable and configure large pages (hugepages) to reduce TLB misses:
   ```
   $ echo 1024 > /proc/sys/vm/nr_hugepages
   $ mount -t hugetlbfs nodev /mnt/huge
   ```
3. Use `numactl` to optimize for the AMD EPYC NUMA architecture by controlling NUMA policy:
   ```
   $ numactl --membind=0 your_application  # Bind to NUMA node 0
   ```
4. Adjust `vm.swappiness` to control swap behavior:
   ```
   $ sysctl -w vm.swappiness=10
   ```
5. Monitor and manage memory usage.
   - Use tools like `free`, `vmstat`, and `sar` to monitor memory usage and swap activity.
   - Implement proper memory management in your applications to avoid memory leaks and inefficient usage.
6. Consider disabling Transparent Huge Pages (THP) for certain latency-sensitive workloads, which can improve performance:
   ```
   $ echo never > /sys/kernel/mm/transparent_hugepage/enabled
   ```

7. Optimize application-specific settings.
    - For databases, adjust buffer pool sizes and caching mechanisms.
    - For big data frameworks like Apache Spark, tune executor memory and other memory-related parameters.
8. Use memory-optimized VM types. For extremely memory-intensive workloads, consider using Microsoft Azure E-series VMs (e.g., Easv6) which offer higher memory-to-vCPU ratios.

2.2.3 Storage Optimizations

IO-intensive workloads are characterized by frequent disk read/write operations that result in high disk I/O activity. These workloads can cause performance bottlenecks because of the relatively slower speed of disk operations compared to CPU and memory operations. Optimizing storage performance is crucial for these workloads. Some examples include:

- Databases (MySQL, PostgreSQL, Oracle)
- Data processing applications
- Virtualized environments
- Video editing/rendering

To optimize I/O performance on AMD EPYC VMs:

1. Use local NVMe SSDs when available
    - The "L" family of VMs are ideal for I/O critical workloads because they come with high throughput, low latency disks directly attached to the VM (up to 19.2 TB total for the largest VM).
    - The "D" and "E" VM families include variants that support locally attached disks up to 880 GB that provide higher IOPS than managed data disks.
    - The "HB" VM family includes 2 x 1.8 TB local disks regardless of the VM size.
2. Use VMs that support Azure premium storage.
    - Ensure that your AMD EPYC VM supports Azure Premium Storage for low latency, high IOPS, and throughput.
    - Enable accelerated networking on the VM to get the best bandwidth between managed data disks and the VM.
3. Choose the right premium storage type
    - Azure Ultra disks support up to 10 GB/s throughput and up to 400,000 IOPS. They are the disk of choice for top tier databases like SAP HANA and Oracle.
    - Azure Premium SSD v2 supports up to 1.2 GB/s throughput and up to 80,000 IOPS. They are good for workloads that require a consistent and high I/O throughput.
    - Azure Premium SSD supports up to 900 MB/s throughput and up to 20,000 IOPS. They are a good choice for many production and performance sensitive workloads.
4. Consider using RAID 0 (striping) across multiple EBS volumes for increased I/O performance:

```
$ mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/xvdf /dev/xvdg
```

5. Optimize the I/O scheduler. For example, for SSDs, use the `noop` or `deadline` scheduler:
   ```
   $ echo noop > /sys/block/nvme0n1/queue/scheduler
   ```

6. Utilize RAM disks for extremely I/O-intensive operations:

   - Create a tmpfs RAM disk:
     ```
     $ sudo mount -t tmpfs -o size=4G tmpfs /mnt/ramdisk
     ```

   - Move frequently accessed data to the RAM disk:
     ```
     $ sudo mv /var/lib/mysql /mnt/ramdisk/
     $ sudo ln -s /mnt/ramdisk/mysql /var/lib/mysql
     ```

7. Monitor I/O performance by using the following tools to identify I/O bottlenecks:

   - **iotop:** Monitor real-time I/O usage of processes.

   - **pidstat:** Print per-process I/O statistics.

   - **iostat:** Monitor system I/O device loading.

   - **vmstat:** Report virtual memory statistics.

8. Tune application-specific settings

   - For databases, adjust buffer pool sizes and I/O-related parameters.

   - For file servers, optimize caching mechanisms and network-related settings.

9. Consider using Azure Filies for shared file systems

   - For workloads requiring shared access across multiple VMs, use Azure Files EFS with the **Max I/O** performance mode.

10. Implement proper I/O patterns in your application"

    - Use asynchronous I/O operations where possible.

    - Implement buffering and caching mechanisms to reduce disk access.

    - Optimize data access patterns to minimize random I/O operations.

2.2.4 Latency Sensitive Optimizations

Latency-sensitive workloads are applications that require low and predictable response times that typically range from microseconds to tens of microseconds. These workloads are often found in areas such as financial trading, online gaming, real-time analytics, and high-performance computing. Optimizing for such workloads involves minimizing sources of latency and variability in the system. Some examples of latency-sensitive workloads include:

- High-frequency trading systems

- Real-time bidding platforms

- Online gaming servers

To optimize latency:

1. Prioritize tasks using `chrt` to identify latency-sensitive tasks and set higher priorities:
   ```
   # Set SCHED_FIFO policy for process with PID 1234 and priority 90
   $ sudo chrt -f -p 90 1234
   # Start a new process with SCHED_RR policy and priority 50
   $ sudo chrt -r -p 50 /path/to/my_latency_sensitive_app
   ```

2. Disable deeper CPU C-states:

   - Install the `cpupower` tool:
     ```
     $ sudo apt install linux-tools-common
     ```

   - Disable C2 state on all cores:
     ```
     $ sudo cpupower idle-set -d 2
     ```

3. Disable Simultaneous Multi-threading (SMT)

   - Check SMT support:
     ```
     $ ls /sys/devices/system/cpu/smt
     ```

   - Disable SMT:
     ```
     $ sudo echo off > /sys/devices/system/cpu/smt/control
     ```

   - Verify SMT is disabled:
     ```
     $ cat /sys/devices/system/cpu/smt/active
     ```

4. Set the maximum core frequency:

   - Check available CPU frequency scaling options:
     ```
     $ cpupower frequency-info
     ```

   - Set the performance governor:
     ```
     $ sudo cpupower frequency-set -g performance
     ```

   - Verify the maximum frequency:
     ```
     $ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
     ```

   - Make the settings persistent:
     ```
     $ sudo systemctl enable cpupower
     $ sudo systemctl start cpupower
     ```

5. Disable Transparent Huge Pages (THP):

   - Disable THP at runtime:
     ```
     $ echo never > /sys/kernel/mm/transparent_hugepage/enabled
     $ echo never > /sys/kernel/mm/transparent_hugepage/defrag
     ```

   - For permanent disabling, add `transparent_hugepage=never` to the kernel boot parameters in the GRUB configuration file.

6. Tune the OS:

   - Use a real-time kernel for critical low-latency applications

   - Adjust kernel parameters for network and I/O performance:
     ```
     bash
     sysctl -w net.core.rmem_max=16777216
     sysctl -w net.core.wmem_max=16777216
     sysctl -w vm.swappiness=0
     ```

7. Apply application-level optimizations:

   - Use lock-free data structures and algorithms where possible.

   - Implement efficient memory management to avoid garbage collection pauses.

   - Use asynchronous I/O operations to prevent blocking on I/O.

8. Monitor and profile your application:

   - Use tools like `perf` and `ftrace` to identify sources of latency in your application.

   - Monitor system-wide latency using `cyclictest` or the `rt-tests` suite.

2.2.5 Operating System Optimizations

Optimizing the operating system is crucial for maximizing the performance of Microsoft Azure VMs powered by AMD EPYC processors. This section covers general OS tuning techniques, with a focus on sysctl parameters and tuned profiles.

2.2.5.1 Sysctl

`Sysctl` allows you to modify kernel parameters at runtime. Here are some recommended `sysctl` settings for optimizing AMD EPYC VMs:

1. Network optimizations:
   ```
   # Increase the maximum number of open file descriptors
   sysctl -w fs.file-max=2097152
   # Increase network buffer sizes
   sysctl -w net.core.rmem_max=16777216
   sysctl -w net.core.wmem_max=16777216
   sysctl -w net.ipv4.tcp_rmem="4096 87380 16777216"
   sysctl -w net.ipv4.tcp_wmem="4096 65536 16777216"
   # Enable TCP Fast Open
   sysctl -w net.ipv4.tcp_fastopen=3
   # Increase the maximum number of incoming connections
   sysctl -w net.core.somaxconn=65535
   ```

2. Virtual memory optimizations:
   ```
   # Reduce swappiness
   sysctl -w vm.swappiness=10
   # Increase the amount of dirty data before writing to disk
   sysctl -w vm.dirty_ratio=60
   sysctl -w vm.dirty_background_ratio=2
   # Optimize for database workloads
   sysctl -w vm.overcommit_memory=2
   sysctl -w vm.overcommit_ratio=95
   ```

3. File system and I/O optimizations:
```
# Increase the maximum number of asynchronous I/O requests
sysctl -w fs.aio-max-nr=1048576
# Optimize for high I/O workloads
sysctl -w vm.dirty_bytes=1073741824
sysctl -w vm.dirty_background_bytes=536870912
```

4. NUMA-specific optimizations:
```
# Enable automatic NUMA balancing
sysctl -w kernel.numa_balancing=1
```

You can make these changes persistent across reboots by either adding them to `/etc/sysctl.conf` or by creating a new file in `/etc/sysctl.d/`.

2.2.5.2 TuneD Profiles

TuneD is a daemon that monitors your system and optimizes its performance under certain workloads and can benefit users who simply want the best "out of the box" optimizations. TuneD includes several profiles (with a few listed below) that optimize your OS for particular applications. You can also create a custom profile for your specific needs.

- **General purpose:** balanced
- **Compute intensive:** throughput-performance
- **Latency sensitive:** latency-performance

Implementing these optimization strategies can significantly enhance the performance of your workloads on Microsoft Azure VMs powered by AMD EPYC processors. Remember to benchmark your specific applications to find the optimal configuration for your use case.

2.3 Cost Considerations

Optimizing costs is a crucial aspect of running workloads on Microsoft Azure VMs. This section presents several strategies for optimizing your costs when running Microsoft Azure VMs powered by AMD EPYC processors.

2.3.1 VM Right-Sizing

VM right-sizing is the process of matching VM types and sizes to your workload performance and capacity requirements at the lowest possible cost. Consider the following for Microsoft Azure VMs powered by AMD EPYC processors:

1. **Analyze current usage:** Use Azure Monitor* to monitor the CPU, memory, network, and disk usage metrics of your current VMs.
2. **Identify underutilized resources:** Look for VMs with consistently low utilization (e.g., below 40% CPU usage) as candidates for downsizing.
3. **Consider workload patterns:** Understand your application's performance requirements and usage patterns over time.
4. **Leverage AMD EPYC advantages:** Microsoft Azure VMs powered by AMD EPYC processors provide optimal price-performance ratios. Learn more at Achieve more with Microsoft Azure and AMD.
5. **Regular review:** Set up a process to regularly review and adjust your VM choices, ideally every 3-6 months or after significant application changes.

2.3.2 Reserved VMs and Savings Plan

You may be able to further optimize the cost of running Microsoft Azure VMs powered by AMD EPYC processors as follows:

- **Reserved VMs:** Reserving VMs can reduce costs compared to on-demand pricing in exchange for a time commitment, such as one or three year(s).
- **Savings Plans:** These can significantly optimize costs in exchange for committing to use individual VM families in a single region.
- **Spot VMs:** Take advantage of unused Microsoft Azure capacity at a potentially significant discount compared to Pay As You Go prices. This may be ideal for fault-tolerant or flexible applications.

Note: AMD does not control the amount and/or applicability of any program and/or cost savings. Please consult Microsoft directly for details.

2.3.3 EPYC VM Advisor Tool

The AMD EPYC VM Advisor (EIA) tool, helps optimize VM selection by helping you:

- Understand performance and cost across your current deployments.
- Make workload-specific recommendations by identifying the most suitable VM type based on your workload characteristics.
- Calculate potential cost savings by switching to Microsoft Azure VMs powered by AMD EPYC processors.
- Deploy quickly and get 1:1 VM recommendations with cost saving estimates.

2.3.4 Cloud Cost Advisor

The AMD Cloud Cost Advisor (CCA) tool offers comprehensive cost optimization estimations:

- Receive suggestions for cost-optimized AMD EPYC VMs that are comparable to your existing deployments.
- Estimate potential savings by migrating to VMs powered by AMD EPYC processors.
- Generate comprehensive reports to help inform decision-making.
- Analyze and optimize costs across multiple cloud providers.

Chapter 3

Monitoring and Observability

Monitoring and observability are crucial aspects of managing Microsoft Azure VMs powered by AMD EPYC processors. These practices help ensure optimal performance, identify bottlenecks, and effective troubleshooting.

3.1 Azure Monitor

Azure Monitor* is the primary monitoring service for Azure resources, including VM VMs powered by AMD EPYC processors. It collects and tracks metrics, which are variables you can measure for your resources and applications. Some key features of Azure Monitor Metrics include:

- Basic and detailed Monitoring:
- Basic monitoring provides data at 5-minute intervals at no charge.
- Detailed monitoring offers data at 1-minute intervals for an additional cost.
- Some available metrics include:
  - CPU Utilization
  - Memory consumption
  - Disk Read/Write Operations
  - Network In/Out
  - Status Check Failed
- **Custom metrics:** You can publish your own custom metrics to Azure Monitor using the Azure CLI or API.
- **Metric math:** Perform mathematical operations on metrics to derive new insights.
- **Metric retention:** Azure Monitor retention period can be configured from days to years.
- **Graphing:** Visualize metrics directly in the Azure Monitor console.
- **Alarms:** Set alarms on metrics to trigger notifications or automated actions.
- **For developers:** Azure Monitor also provide tools to help developers analyze and debug distributed applications, providing insights into application performance and behavior.

3.2 Resource Monitoring Tools

Several tools can help monitor the performance of AMD EPYC VMs:

- **Azure Portal Dashboard:** Provides a high-level overview of your VMs, including those powered by AMD EPYC processors.
- **Azure Automation:** Offers a unified user interface to view operational data from multiple Azure services and automate tasks across your Azure resources.
- **Third-party monitoring solutions:**
  - **Datadog:** Offers comprehensive monitoring for cloud environments, including specific features for AMD EPYC VMs. Learn more here*.
  - **Dynatrace:** Offers AI-powered, full-stack monitoring with specific integrations for Microsoft Azure services. Lean more here*.
- Open-source tools:
  - **Prometheus:** A popular open-source monitoring and alerting toolkit. Learn more here*.
  - **Grafana:** An open-source platform for monitoring and observability, often used in conjunction with Prometheus. Learn more here*.

3.3 VM Profiling Tools

Profiling tools help identify performance bottlenecks and optimize code execution on AMD EPYC VMs:

- Virtual Memory Statistics (`vmstat`) provides information about system processes, memory, paging, block I/O, traps, and CPU activity.
- Input/Output Statistics (`iostat`) reports CPU statistics and input/output statistics for devices and partitions.
- Network Statistics (`netstat`) provides information about network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.
- Linux profiling with performance counters (`perf`) is a powerful Linux profiling tool that provides detailed CPU performance analysis.
- `Valgrind` is an instrumentation framework for building dynamic analysis tools, useful for memory debugging and profiling.
- AMD uProf is AMD's proprietary profiling tool for detailed performance analysis of AMD processors.
- `eBPF` (extended Berkeley Packet Filter) is a powerful and flexible Linux kernel technology that can be used for performance analysis and monitoring.

Consider the following best practices when using these tools:

- Profile in production-like environments to get accurate results.
- Focus on hot paths and frequently executed code.
- Use a combination of tools to get a comprehensive view of performance.
- Regularly profile your applications to catch performance regressions early.

This page intentionally left blank.

Chapter 4

Troubleshooting and Support

Leveraging the support resources and best practices described in this chapter will help you effectively troubleshoot and resolve issues with Microsoft Azure VMs powered by AMD EPYC processors, thereby ensuring optimal workload performance and reliability.

4.1 Noisy Neighbor Mitigation

The “noisy neighbor” is a common issue with cloud VMs where running multiple VMs on the same physical host can impact the performance of other VMs on the same physical host. This may be noticeable with memory bandwidth on AMD EPYC processors due to their multi-CCD design. To check if you're experiencing noisy neighbor effects on memory bandwidth:

1. Run the STREAM benchmark to measure memory bandwidth.
2. Compare your results to the expected bandwidth for your VM type. For example, a Microsoft Azure Standard_F8as_v6 VM placed on a single CCD should achieve around 50 GB/s.

If your bandwidth is significantly lower, you may be experiencing noisy neighbor effects. To resolve this issue:

- Request a new VM and test again. Keep requesting new VMs until you get one with expected performance.
- Consider using dedicated VMs for consistent performance.

4.2 VM Placement Verification

The VM vCPUs might not be ideally placed across CCDs, which can impact performance for some workloads. To verify your VM placement:

1. Use the core-to-core latency tool*.
2. Analyze the output. You should see lower latencies between cores on the same CCD.

If you see unexpectedly high latencies between cores that should be on the same CCD, then your VM may be poorly placed. To resolve this issue:

- Request a new VM and test again until you get one with optimal core placement.
- For critical workloads, consider using dedicated hosts to ensure consistent placement.

4.3 Auto Scaling and Load Balancing

Issues with auto scaling and load balancing can lead to performance problems and increased costs. Common issues include:

- Scaling too slowly or quickly in response to demand.
- Uneven load distribution across VMs.
- Scaling based on inappropriate metrics.

Some solutions and best practices include:

- **Use appropriate scaling metrics:** Choose metrics that directly correlate with your application's performance, such as request latency or queue length, rather than just CPU utilization.
- **Set appropriate scaling thresholds:** Configure scaling policies to react quickly enough to demand changes without causing oscillation.
- **Implement proper health checks:** Ensure your load balancer and auto scaling group use appropriate health checks to detect and replace unhealthy VMs.
- **Use target tracking scaling policies:** These policies automatically adjust capacity to maintain a specific metric at a target value.
- **Implement gradual scaling:** Use step scaling policies to add or remove capacity in increments based on alarm breach size.
- **Optimize VM warm-up:** Set appropriate cooldown periods and health check grace periods to allow new VMs to warm up before receiving traffic.
- **Use multiple Availability Zones:** Distribute your Auto Scaling group across multiple AZs for better fault tolerance and performance.
- **Consider using Spot VMs:** For flexible workloads, use a mix of On-Demand and Spot VMs to optimize costs.
- **Implement proper application-level load balancing:** Ensure your application can distribute work evenly across VMs, especially for stateful workloads.
- **Monitor and adjust:** Regularly review your auto scaling and load balancing performance using Azure Monitor and adjust settings as needed.

4.4 Azure Support Resources

- Azure Documentation* offers comprehensive guides and best practices for AMD VMs.
- Azure Support* is the primary hub for creating and managing support cases. Available to all Azure customers, with response times varying by support plan.
- Azure Advisor* provides free guidance to help you provision your resources following Azure best practices. This includes recommendations for VMs powered by AMD EPYC processors.
- Azure Service Health* offers personalized views of Azure service health, including any issues affecting your AMD VMs.
- Azure Support API* delivers programmatic access to Azure Support features that allows you to integrate support workflows into your applications.

4.5 AMD Support Resources

In addition to Microsoft support, AMD provides resources specifically for EPYC processors:

- AMD Developer Central offers optimization guides, technical documentation, and best practices for AMD EPYC processors.
- AMD Enterprise Support provides direct support channels for hardware-specific issues to enterprise customers.
- AMD Support Forum is a platform where developers and system administrators can discuss AMD-specific topics and share solutions.
- AMD works with various software vendors across the AMD Data Center Partner Ecosystem to ensure compatibility and optimization. Check with your software provider for AMD-specific support.

AMD recommends using Microsoft support channels first when troubleshooting issues with Microsoft Azure VMs powered by AMD EPYC processors. If the issue is determined to be specific to the AMD processor architecture, then you may need to engage AMD's support resources.

Remember to provide detailed information when seeking support, including:

- VM type and OS and configurations details
- Exact error messages or symptoms
- Steps to reproduce the issue
- Any recent changes to your environment
- Relevant Azure Monitor metrics or logs

This page intentionally left blank.