Understanding Capacities

Andrea Benedetti
Sr Cloud Architect, Microsoft
/in/abenedetti @anBenedetti https://github.com/anbened

#FabricGarage | 005 | 2024.06.13

What are Capacities?

Everything you need to know about

- Microsoft Fabric = unified data platform
  - Shared experiences / architecture / governance / compliance / billing

- Capacities = foundation of the Microsoft Fabric platform
  - Providing the compute resources that power all the experiences in the platform
  - Will run workloads concurrently and don't need to be specifically allocated to a resource

- Each capacity is a dedicated set of resources that is reserved for exclusive use

What are Capacities?

Everything you need to know about

- Capacities provide the computing power that drives all of these experiences
  - A simple and unified way to scale resources (to meet customer demand)
  - Can be easily increased with a SKU upgrade
  - One capacity to drive all your Fabric experiences

What are Capacities?

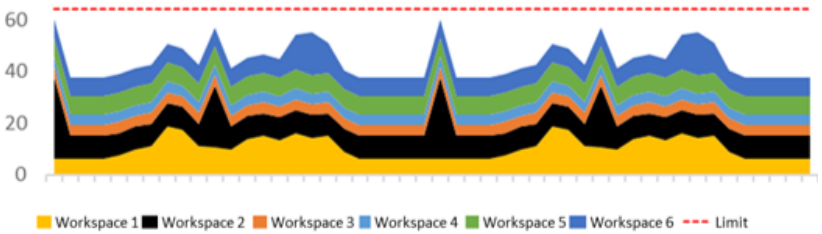
Everything you need to know about

- Capacities are to Fabric what CPUs are to PCs
  - When you buy a PC, you think about the number of CPU cores you want to purchase. Is it a PC for lightweight business use? Or gaming and video production…
  - PC
    - The CPU cores are dynamically shared across all applications with no need to pre-allocate by app
    - The total consumption of the CPU across all the apps cannot exceed the number of cores. CPU overload causes a slowdown
  - Fabric
    - The capacity units are dynamically shared across all the Fabric workloads, with no pre-allocation necessary
    - A single capacity can simultaneously drive BI, DW, ML, … and every other compute engine in Fabric
    - The total consumption of the capacity across all the workloads cannot exceed the capacity units provisioned
    - Overloading the capacity will throttle it (slow down)
    - Auto scale can dynamically increase the available compute units avoiding the slowdown

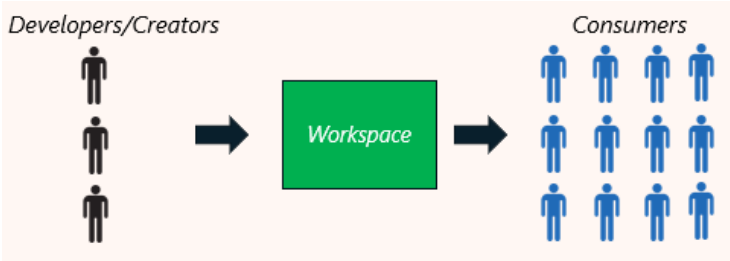
Capacities are a shared resource...

- ... across workloads

- ... across Projects



- ... across users



Capacities are a SaaS resource...

- Always on auto-pilot
  - A self-managed system
  - Providing complete transparency into the usage
    - projects/workspaces, artifacts, queries, users and jobs
- Eliminate workload management
  - Large jobs can be scheduled at any time without impacting any other jobs running
  - Protection against a single user
    - hogging all the resources by carelessly running a very large query
  - Smoothing automatically prevents spikey loads from creating temporal overload of the system
- Always peak performance
  - Bursting allows all jobs and queries to always run at peak performance
  - Auto scale can automatically adjust the compute units provisioned to the capacity

CUs

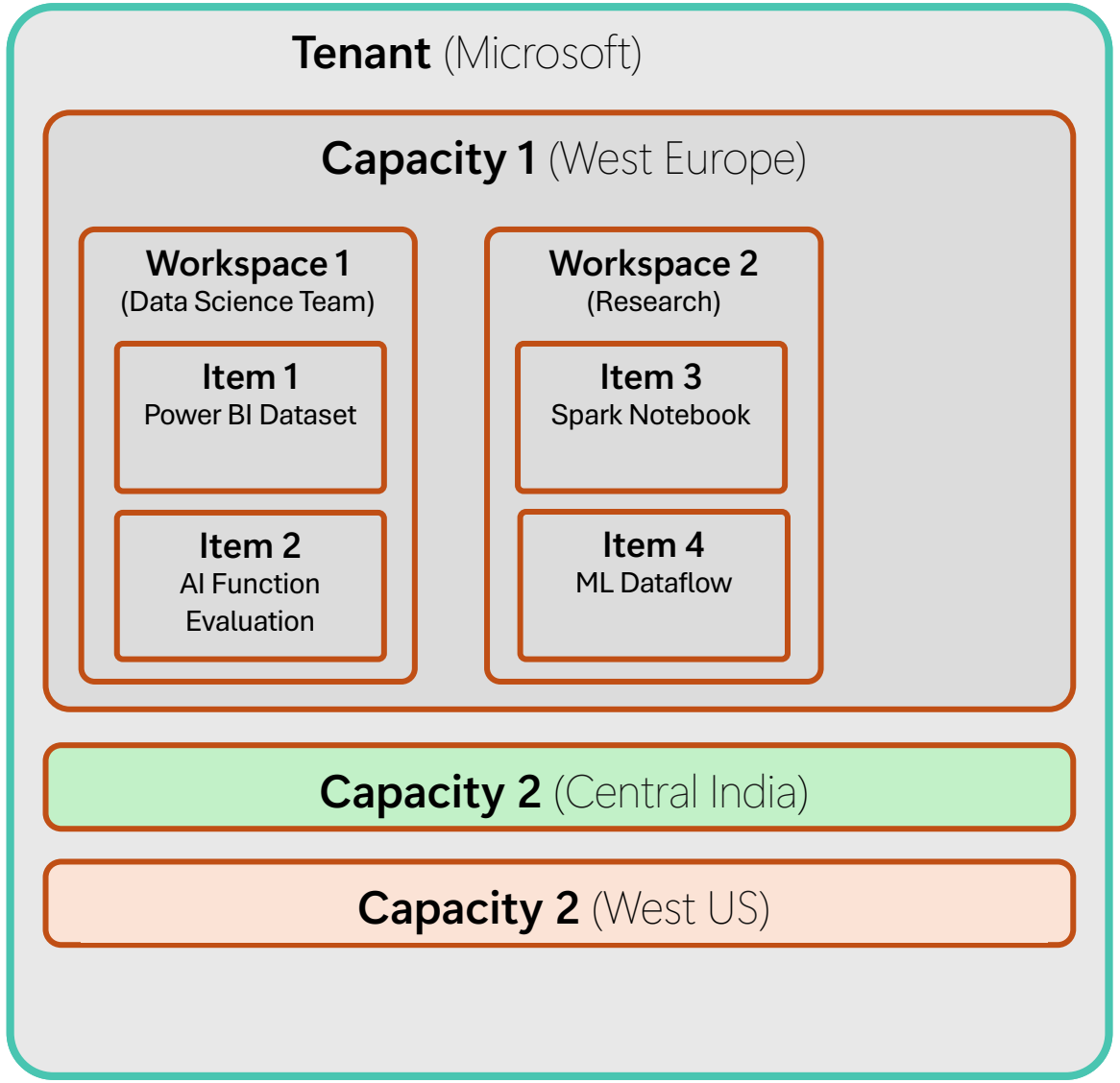
- Capacity units (CUs) = the metric to represent capacity allocated
  - Options range: minimum of 2 CUs → maximum of 2048 Cus

SKU	Capacity Units (CU)	Power BI SKU	Power BI v-cores
F2	2	-	0.25
F4	4	-	0.5
F8	8	EM/A1	1
F16	16	EM2/A2	2
F32	32	EM3/A3	4
F64	64	P1/A4	8
F128	128	P2/A5	16
F256	256	P3/A6	32
F512	512	P4/A7	64
F1024	1024	P5/A8	128
F2048	2048	-	256

- Key innovation: efficient and effective management of data solutions
  - Users can leverage full cloud potential / no intricate resource management strategies

Provisioning and Deploying Capacities

- Capacity = specific region
- Workspaces assigned to a capacity
- Multiple capacities:
  - can be purchased, deployed and managed by different owners residing in a single tenant allowing each business unit to pay for their own consumption



Bursting

Bursting for blazing performance running Fabric experiences

- Bursting allows you to consume extra compute resources beyond what have been purchased to speed the execution of a workload
  - If you need more power for a short amount of time, Fabric can ‘burst’ your compute power above the limit you’ve bought
  - For example, instead of running a job on 64 CU and completing in 60 seconds, bursting could use 256 CUs to complete the job in 15 seconds
- Bursting
  - SaaS feature and requires no user management
  - Designed to enhance performance & stability
    - by allowing workloads to access more resources than their allocated baseline capacity
- Compute spikes generated from bursting will not cause throttling due to smoothing policies

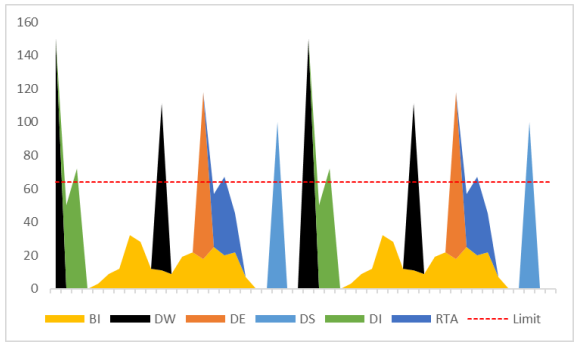
Smoothing

Smoothing helps streamline management by allowing you to plan for average usage instead of peak

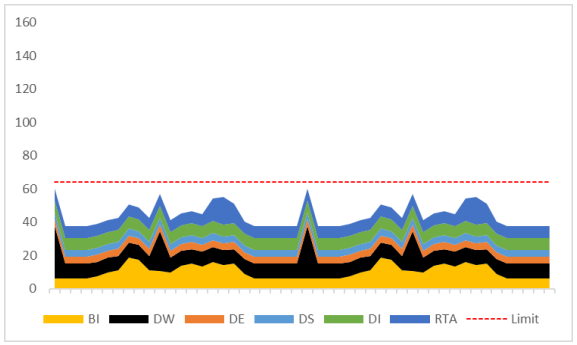
- When a capacity is running multiple jobs, a sudden spike in compute demand may be generated that exceeds the limits of a purchased capacity.
- Smoothing simplifies capacity management here by spreading the evaluation of compute to ensure that your jobs run smoothly and efficiently.
- Smoothing simply also allows you to size your capacity based on average, not peak usage

Background operations (such as scheduled data refreshes or data pipeline processes that run without immediate user interaction) are smoothed over time, distributing execution to maintain system performance without exceeding capacity

Before Smoothing



After Smoothing



Andrea Benedetti
Sr Cloud Architect, Microsoft
/in/abenedetti @anBenedetti https://github.com/anbened

#FabricGarage | 005 | 2024.06.13