

# SQL Server 2022 Degree of Parallelism (DOP) Feedback

Sergio Govoni

Slide and demo: <https://bit.ly/42MQpjo>

# Speaker bio



Sergio Govoni



[twitter.com/segovoni](https://twitter.com/segovoni)



[github.com/segovoni](https://github.com/segovoni)



[linkedin.com/in/sgovoni](https://linkedin.com/in/sgovoni)

# Agenda

- Introduction to
  - Parallel processing and CXPACKET
  - Max degree of parallelism (MAXDOP)
  - SQL Server 2022 degree of parallelism (DOP) feedback



# Parallel processing

CXPACKET

# What is parallel processing?

- Parallel processing is a technique that aims to separate big tasks into more than one small task, and these small tasks will be completed by the discrete threads
- SQL Server tries to process queries that require excessive workload in a parallel manner
- Designed to be automatically
- One thread per set of rows



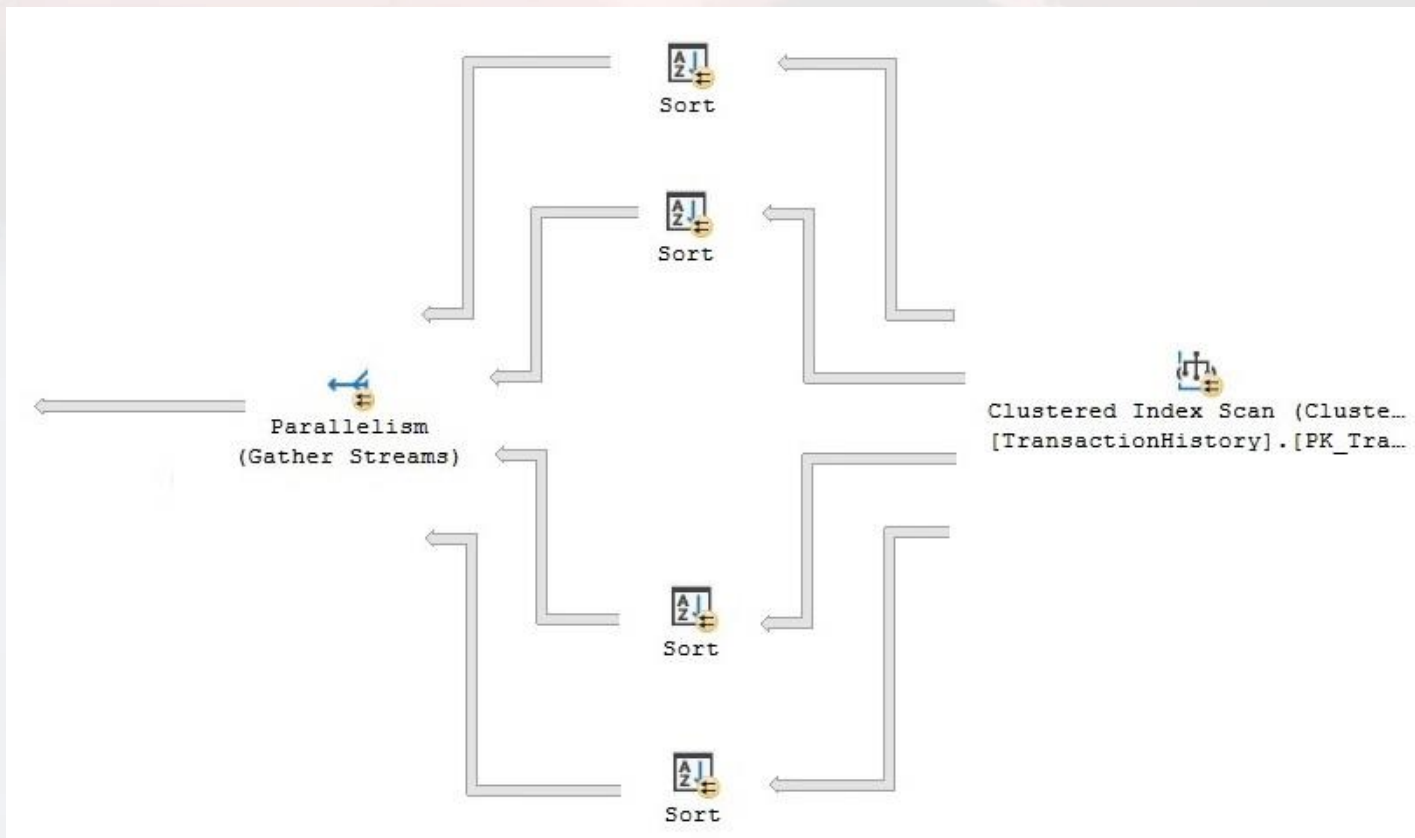
# What is parallel processing?

- SQL Server Query Optimizer considers three settings when generating a parallel query plan
  - Cost Threshold for Parallelism
  - Max Degree of Parallelism (MAXDOP)
  - Affinity mask

# SQL Server and parallel processing

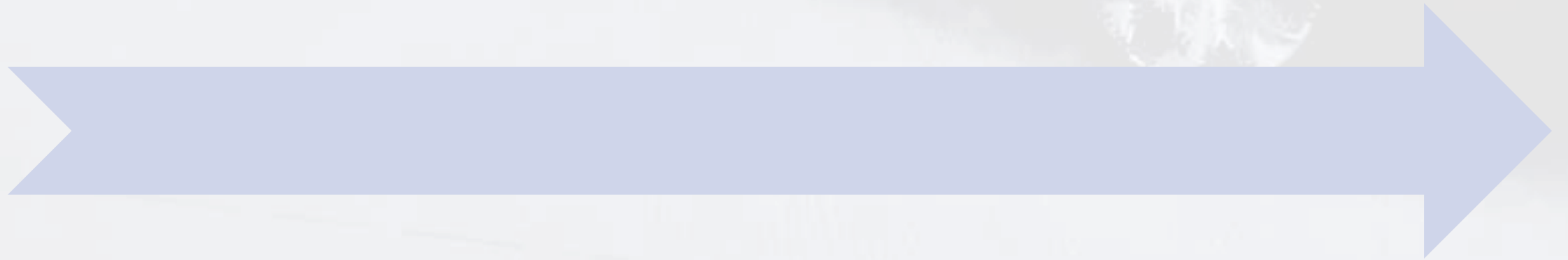
- SQL Server considers parallel execution plans for
  - Queries
  - Index
  - Data definition language (DDL) operations
  - Parallel inserts
  - Online alter column
  - Parallel stats collection
  - Static and keyset-driven cursor population

What really happens?





How it really works?



# Amdahl's law

Amdahl's law is often used in parallel computing to predict the theoretical speedup when using multiple processors

$$\% \text{ improvement} = \frac{1}{(1 - p) + \frac{p}{n}}$$

p = Percent of an algorithm that can be parallelized

n = Number of processor on which the algorithm will be parallelized



**Gene Amdahl**

Computer architect and technology entrepreneur

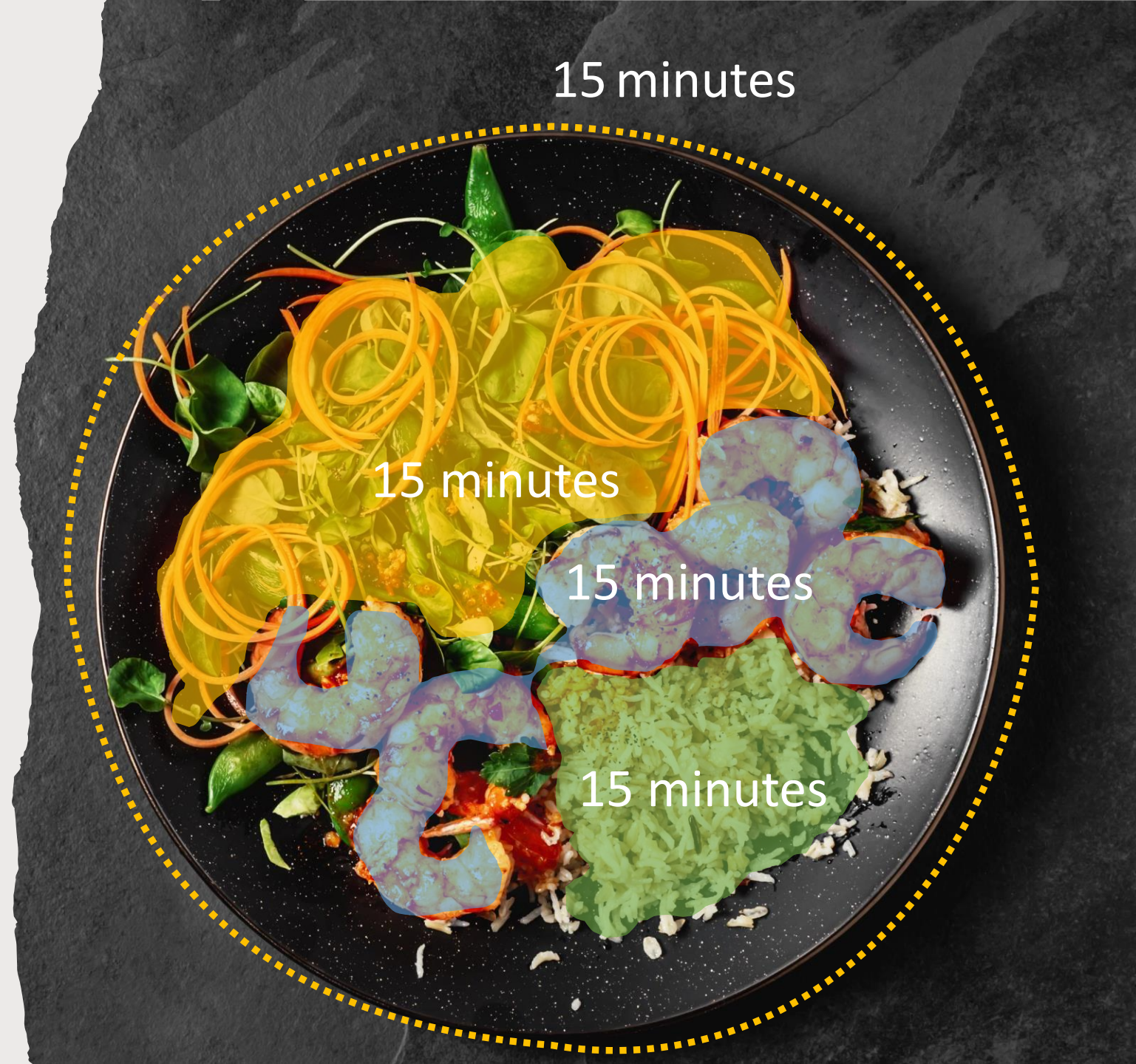
Dish of the  
day





# Dish of the day

$$\% \textit{ improvement} = \frac{1}{(1 - p) + \frac{p}{n}}$$





Hire 3 cooks, and...

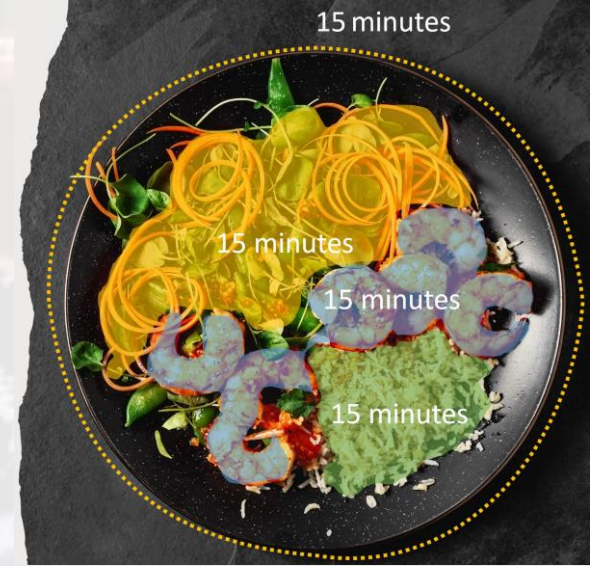
# Amdahl's law in practice

3 of 4 parts are parallelizable,  $p = 0,75$

Each cook is a parallel “core”,  $n = 3$

$$\frac{1}{(1 - p) + \frac{p}{n}} = \frac{1}{(1 - 0,75) + \frac{0,75}{3}} = \frac{1}{0,25 + 0,25} = \frac{1}{0,50} = 2$$

With 3 cooks the preparation time of the dish drops from 60 to 30 minutes, with a 50% improvement!





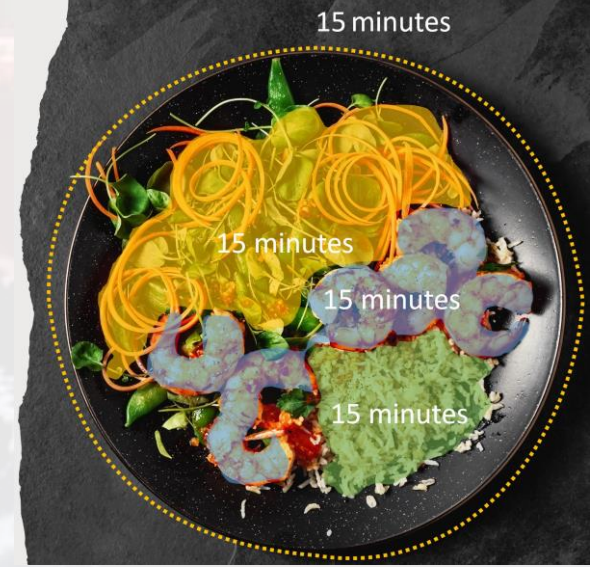


Hire 6 cooks, and...

# Amdahl's law in practice

3 of 4 parts are parallelizable,  $p = 0,75$

Each cook is a parallel “core”,  $n = 6$



$$\frac{1}{(1 - p) + \frac{p}{n}} = \frac{1}{(1 - 0,75) + \frac{0,75}{6}} = \frac{1}{0,25 + 0,125} = \frac{1}{0,375} = 2,66$$

With 3 cooks the preparation time of the dish drops from 60 to 30 minutes, with a 37,5% improvement!

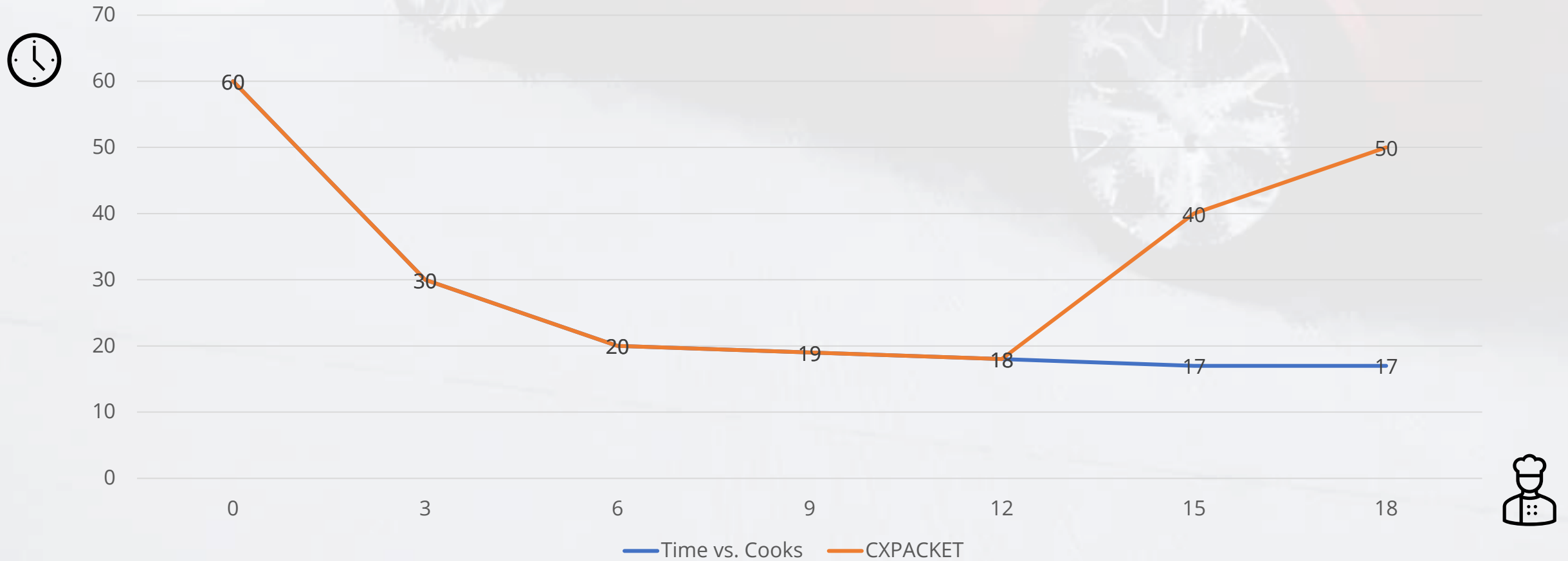
# Amdahl's law in practice



Time vs. Cooks



# Amdahl's law in practice





CXPACKET

# CXPACKET

- What exactly is CXPACKET?
  - Class **EX**change **P**ackets
- It occurs with Parallel Query Plans when trying to synchronize the query processor exchange iterator
- Parallelism issue through Parallel Execution Plan
- Parallel threads are not given equal amount of work to do

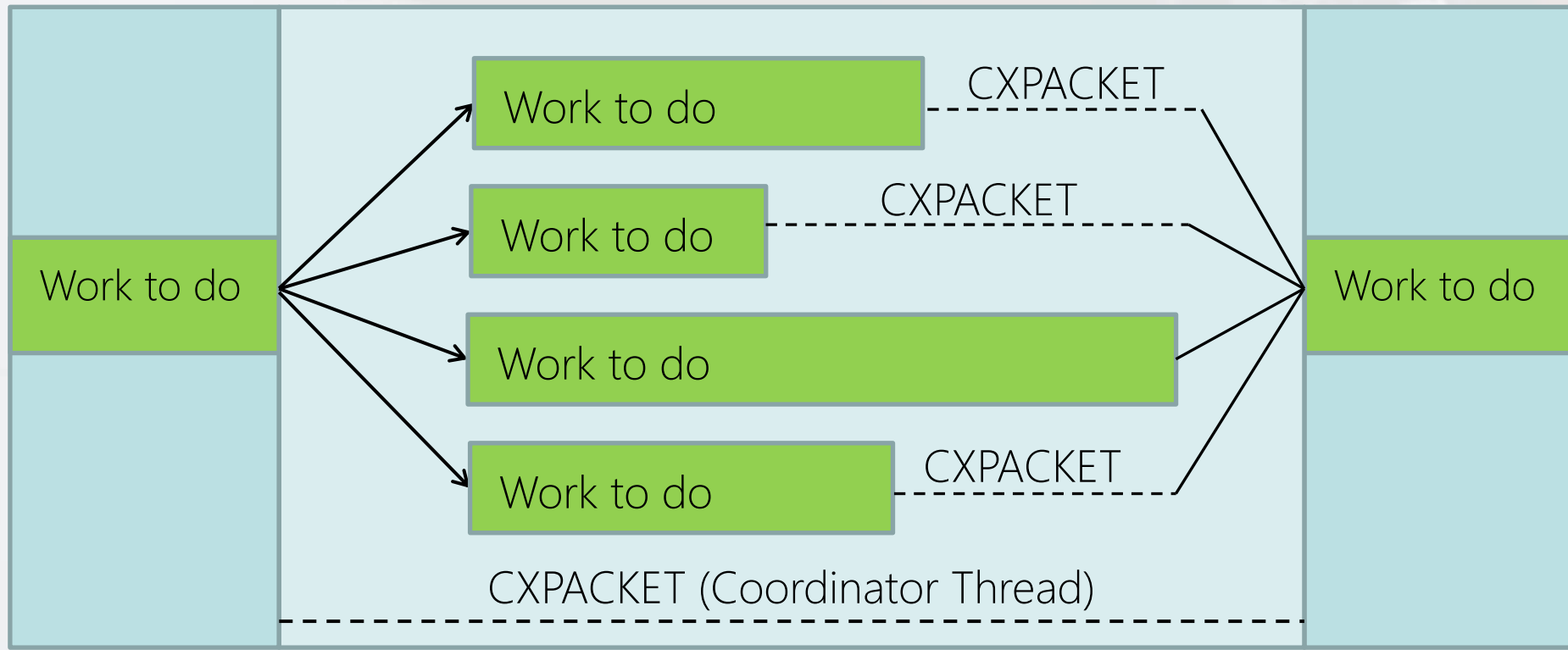


# CXPACKET

Single Thread

Multi Thread

Single Thread





DEMO



# **MAXDOP**

Maximum degree of parallelism

# What is max degree of parallelism?

- The **maximum degree of parallelism** (MAXDOP) is a configuration option that controls the number of **processor** used to run a single statement
- When an instance of SQL Server runs on a computer that has more than one processor or CPU, the Database Engine detects whether parallelism can be used

# What is max degree of parallelism?

- The degree of parallelism sets the number of processors employed to run a single statement, for each **parallel plan execution**
- You can use the max degree of parallelism (MAXDOP) option to limit the number of processors to use in parallel plan execution

# MAXDOP: Before you begin

- MAXDOP is an advanced option, and it should be changed only by an experienced DBA
- Setting MAXDOP to 0 allows SQL Server to use all the available processors up to 64 processors. However, this is not the recommended value for most cases
- You can disable parallel plan setting MAXDOP to 1. However, this is not the recommended value for most cases



# MAXDOP: Before you begin

- Set the MAXDOP to a number from 1 to 32.767 to specify the maximum number of processor cores that can be used
- The max degree of parallelism limit is set per [task](#). It is not per [request](#) or per query limit
- During a parallel query execution, a single request can spawn multiple tasks up to the MAXDOP limit, and each task will use one worker and one scheduler

# MAXDOP: Before you begin

- For requests that are executed in parallel, you will see multiple rows in [sys.dm\\_os\\_tasks](#) for the same combination of session\_id, request\_id
- If the affinity mask option is not set to the default, it may restrict the number of processors available to SQL Server on symmetric multiprocessing (SMP) systems

# MAXDOP configuration

- If you are an experienced DBA 😊 you can override the max degree of parallelism server configuration value in this ways:
  - At the query level, using the MAXDOP [query hint](#) or [Query Store hint](#)
  - At the database level, using the MAXDOP [database scoped configuration](#)
  - At the workload level, using the MAX\_DOP [Resource Governor workload group configuration option](#)
  - At the index statement level, using MAXDOP index option [Configure Parallel Index Operations](#)
  - At the DBCC CHECKTABLE, DBCC CHECKDB, and DBCC CHECKFILEGROUP level, you can disable parallel execution plans for these statements by using [trace flag 2528](#)

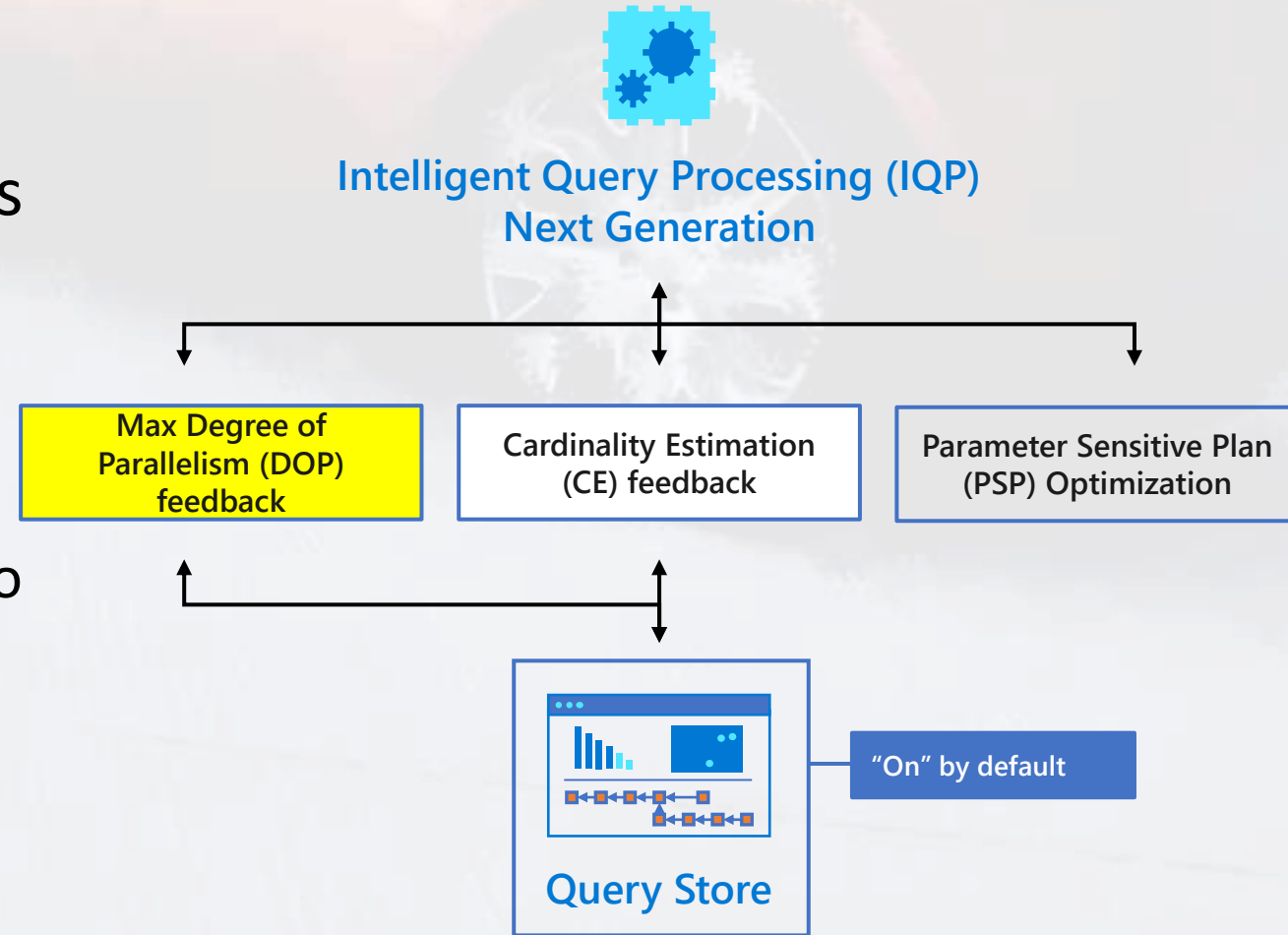


# **SQL Server 2022 DOP feedback**

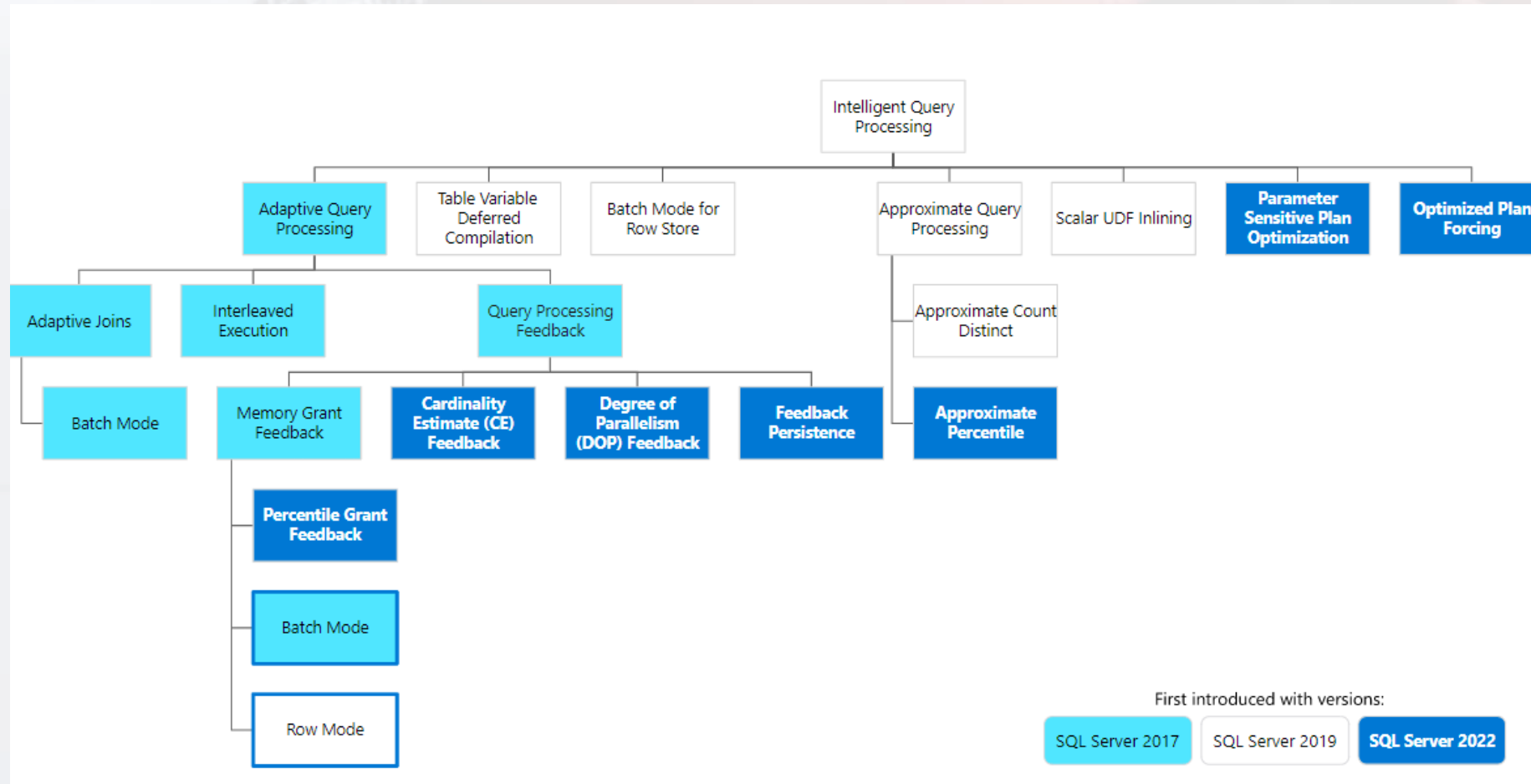
The definitive solution to the MAXDOP configuration!

# Query Store and Intelligent Query Processing

- Accelerate query performance and tuning with no code changes
  - Query Store now turned on by default
  - Query Store support for read replicas from availability groups
  - Query hints to shape plans with no code changes
  - New IQP scenarios enabled through better together capabilities



# Intelligent Query Processing





# Degree of Parallelism (DOP) Feedback

- DOP feedback is one of the new features of SQL Server 2022
- It is part of the family features known as [Intelligent Query Processing](#) that improve the performance of existing workloads without changes to the application code
- It addressing the scenario that occurs when an **OLTP query** is repeatedly executed in **parallel mode** and **performance issues** are encountered

# Degree of Parallelism (DOP) Feedback

- DOP feedback will identify for you the **inefficiencies due to parallelism** for repeated queries, based on **query elapsed time** and **waits**
- If the use of parallelism is deemed inefficient, DOP feedback will **reduce** the degree of parallelism in the next execution of the query

# Degree of Parallelism (DOP) Feedback

- The goal of DOP feedback is to **increase overall concurrency** and **reduce waits**
- Only **verified** feedbacks will be kept
- A stable correction of the DOP is checked at each recompilation of the execution plan and can may **readjust**

# Monitoring the DOP feedback

- It is possible to monitor the functioning of DOP feedback through [sys.query\\_store\\_plan\\_feedback](#) and [6 new Extended Events](#)
  - dop\_feedback\_eligible\_query
  - dop\_feedback\_provided
  - dop\_feedback\_validation
  - dop\_feedback\_stabilized
  - dop\_feedback\_reverted
  - dop\_feedback\_analysis\_stopped

Demo



# Resources

- [SQL Server 2022 Degree of Parallelism \(DOP\) Feedback](#)
- [Configure the max degree of parallelism Server Configuration Option](#)
- [Configure the max degree of parallelism \(MAXDOP\) in Azure SQL Database](#)
- [Best practices for deploying SQL Server on Amazon EC2 - Set MAXDOP for best performance](#)
- [Intelligent Query Processing \(IQP\)](#)
- [Azure SQL & SQL Server 2022: Intelligent Database Futures](#)

# Summary

Degree of parallelism (DOP) feedback is one of the new features of SQL Server 2022 and it is part of the Intelligent Query Processing (IQP) family features

It can **self-adjust** the MAXDOP option to avoid performance problems due to the use of parallelism in OLTP queries performed repeatedly

# Thanks

Questions?



[github.com/segovoni](https://github.com/segovoni)



[twitter.com/segovoni](https://twitter.com/segovoni)



[linkedin.com/in/sgovoni](https://linkedin.com/in/sgovoni)