

# Azure Data Lake Analytics

*Manjunath Suryanarayana*



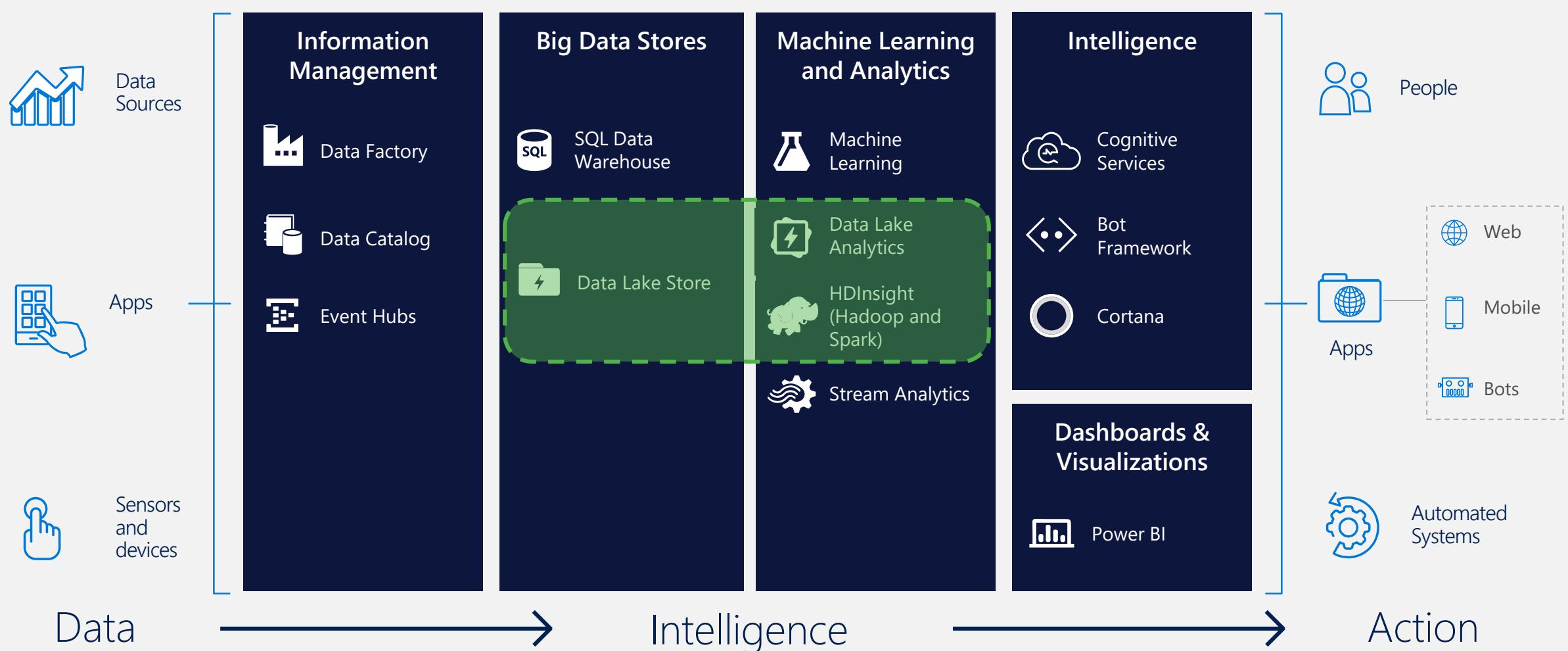
10101010101110111011  
1010101010101010111010  
10101010101110111011

1010111010 101011101  
10111010 101011101  
1111011 1101101  
10101010 101011101  
10101010 101011101

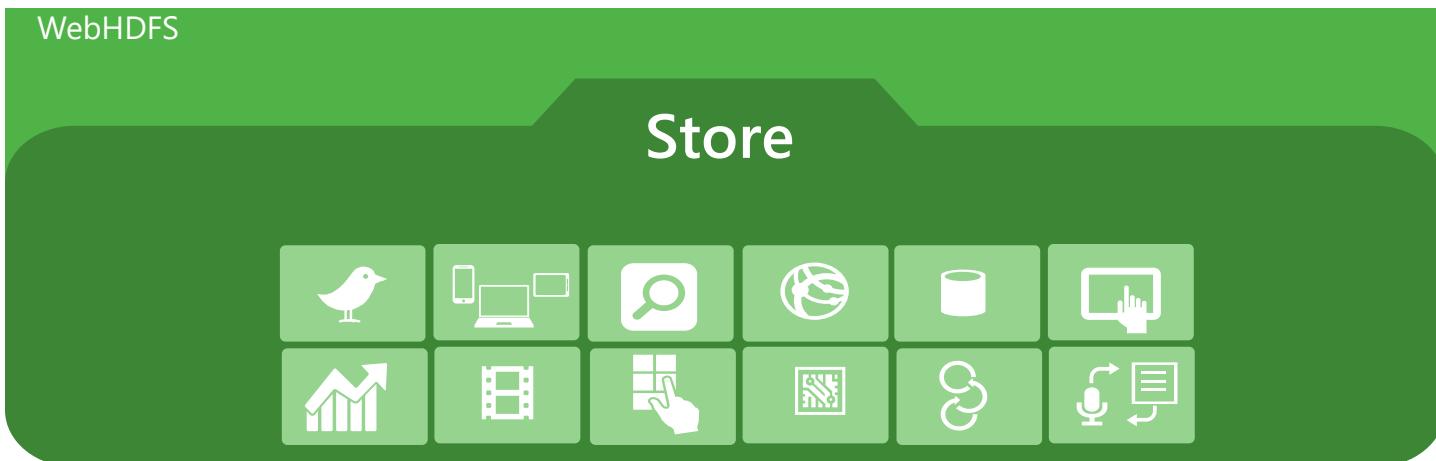
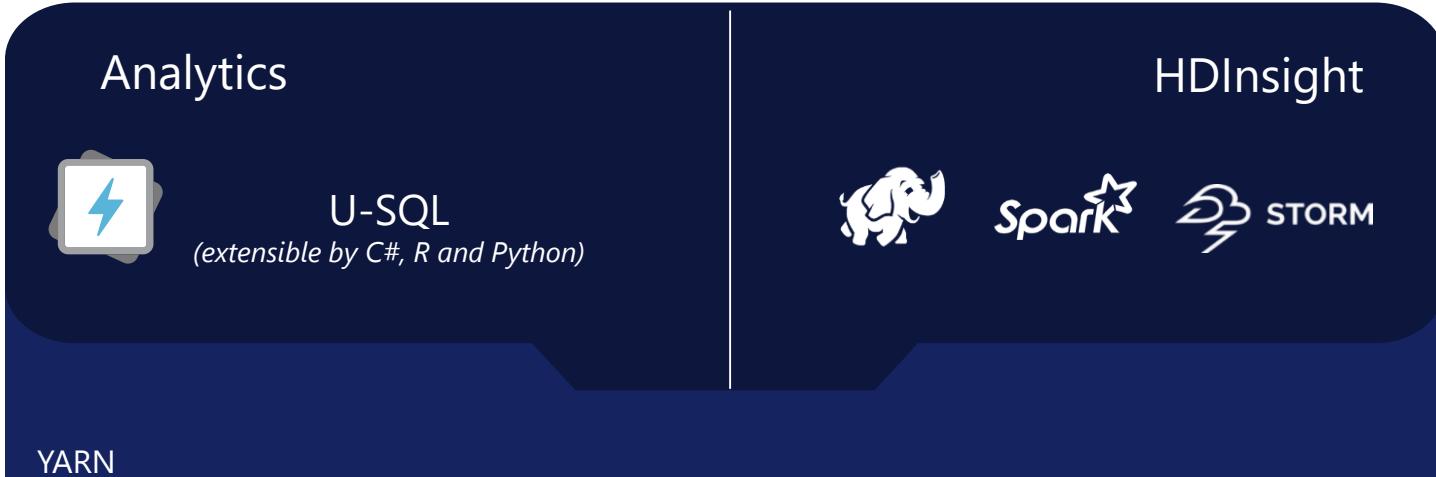
1010111010 101011101  
10111010 101011101  
1111011 1101101  
10101010 101011101  
10101010 101011101



# Azure Data Lake as part of Cortana Intelligence Suite



# Azure Data Lake



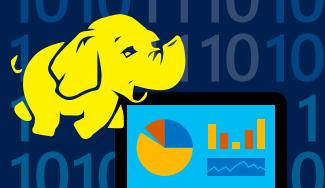
# Azure Data Lake Analytics



10101010101110111011  
1010101010101010111010  
10101010101110111011

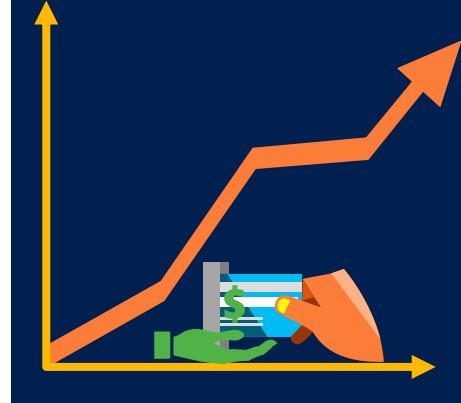
1010111010 101011101  
10111010 101011101  
1111011 1110111  
10101010 101011101  
10101010 101011101

1010111010 101011101  
10111010 101011101  
1111011 1110111  
10101010 101011101  
10101010 101011101

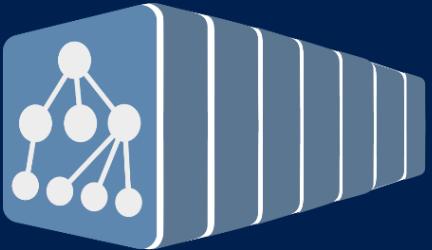


# Azure Data Lake Analytics

Start in seconds  
Scale instantly  
Pay per job



Develop massively parallel programs with simplicity



Debug and optimize your Big Data programs with ease



Virtualize your analytics



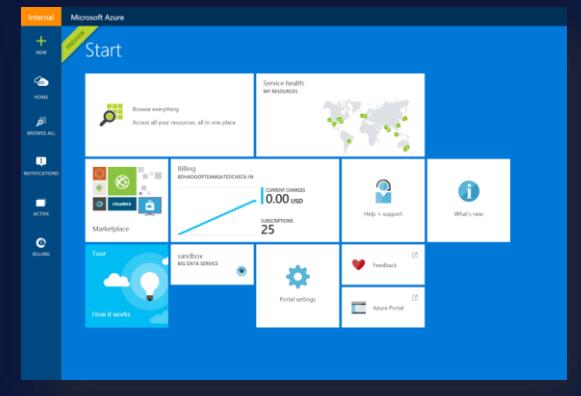
Enterprise-grade security, auditing and support



# Get started

1

Log in to Azure



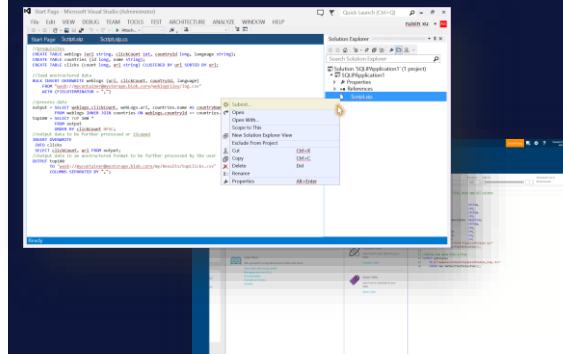
2

Create an ADLA account

30 seconds

3

Write and submit an ADLA job with U-SQL



4

The job reads and writes data from storage

**ADLS**  
**Azure Blobs**  
**Azure DB**  
...

# What can you do in the Azure Portal?

 Create a new Data Lake Analytics account

 Author U-SQL scripts

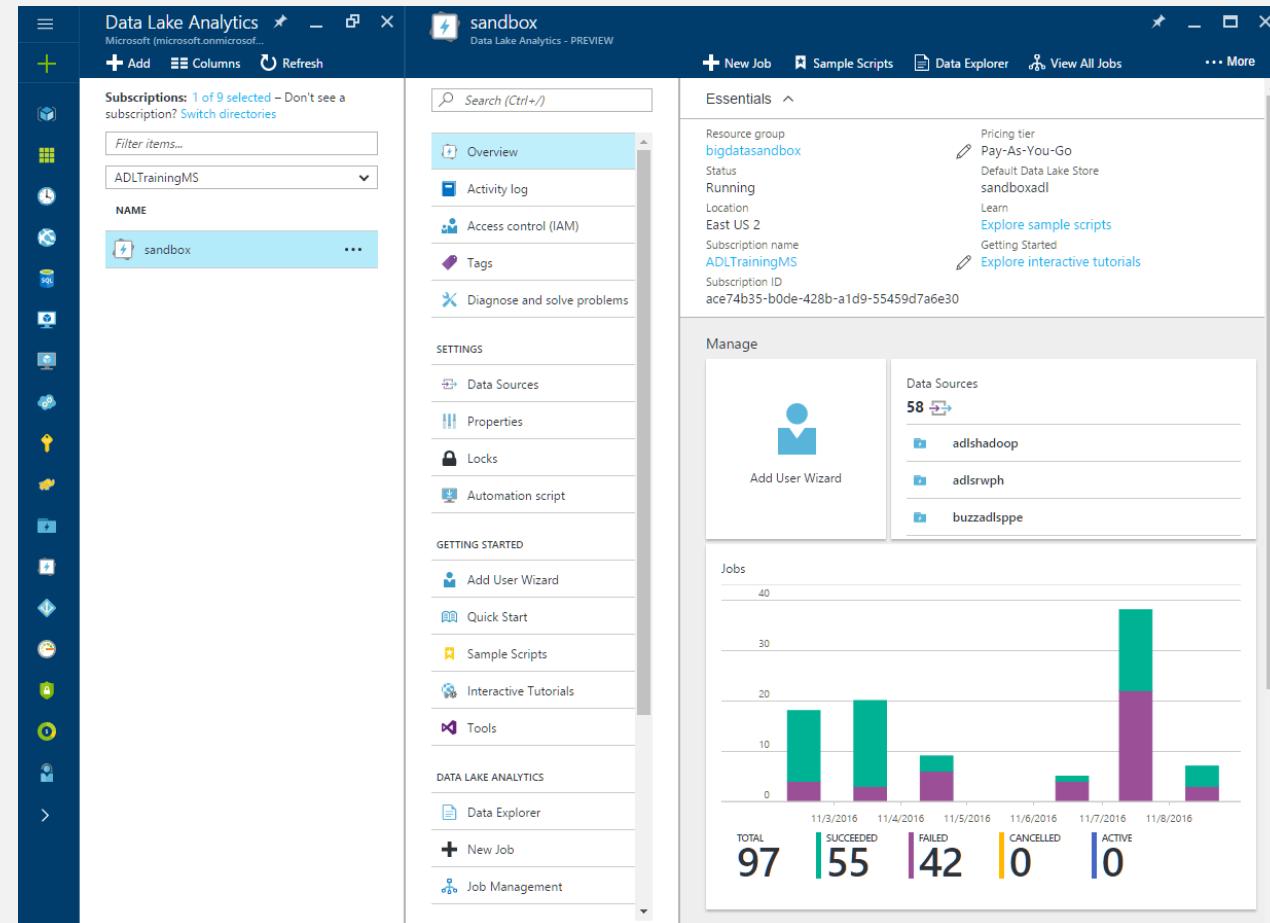
 Submit U-SQL jobs

 Cancel running jobs

 Provision users who can submit jobs

 Visualize usage stats (compute hours)

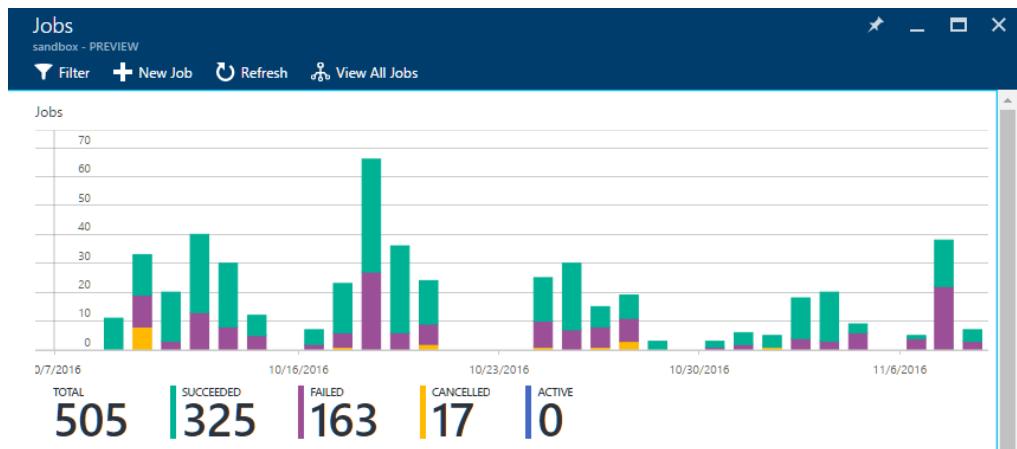
 Visualize job management chart



# Visualizing ADL Analytics monitoring charts

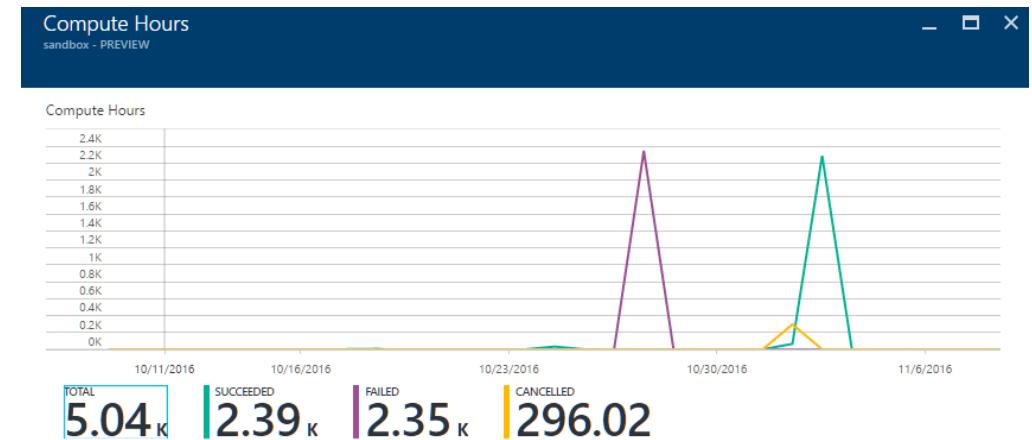
## 1. Job management:

The total number of jobs submitted as well as the number that succeeded, failed or were cancelled



## 2. Usage:

The number of compute hours consumed by the jobs



# Creating an ADL Analytics account

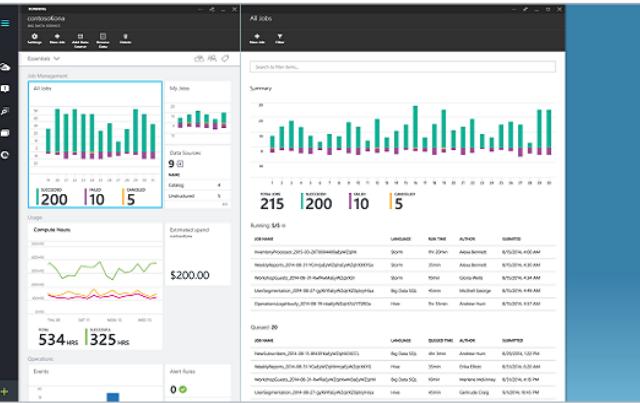
 Data Lake Analytics (preview)  
Microsoft

The Azure Data Lake Analytics service was architected from the ground up for cloud scale and performance. It takes away the complexities normally associated with big data in the cloud and ensures that Data Lake Analytics will meet your current and future business needs.

Highlights:

- Analyze any kind of data of any size
- Only pay for the processing power that you use
- Develop faster, debug and optimize smarter
- Introducing U-SQL: simple, familiar, and extensible
- Managed and supported with an enterprise-grade SLA
- Dynamically scales to match your business priorities
- Built on YARN, designed for the cloud
- Proven at Microsoft with more than 10,000 developers





PUBLISHER Microsoft

New Data Lake Analyti... PREVIEW

Name adldemoaccount ✓  
Subscription Azure conversion - Internal ✓  
Resource Group \* Create new Use existing adldemo  
Location East US 2

\* Data Lake Store Configure required settings >

Pricing Tier Pay-As-You-Go

Pin to dashboard

**Create Automation options**

Select Data Lake Store PREVIEW

Subscriptions: 1 of 9 selected – Don't see a subscription? Switch directories

Filter items... ADLTrainingMS

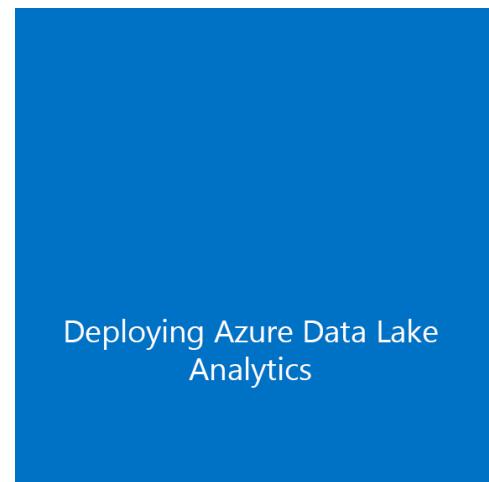
+ Create New Data Lake Store

sandboxadl East US 2

New Data Lake Store PREVIEW

Name adldemoaccountadls  
Subscription adldemoaccountadls.azuredatalakestore.net  
Pricing Pay-As-You-Go  
Encryption Settings Enabled

• Create a new ADL store default account, or  
• Associate with an existing ADL store account (with the right permissions)



# Provisioning users for Data Lake Analytics

Users have to be explicitly provisioned for both the Big Data Lake account as well as for the associated ADL Store account

The screenshot shows the Azure portal interface for provisioning a user for Data Lake Analytics. On the left, the 'Overview' tab is selected in the main navigation bar. A 'Resource group' card displays details like 'bigdatasandbox', 'Status: Running', 'Location: East US 2', and 'Pricing tier: Pay-As-You-Go'. In the center, the 'Add User Wizard' window is open, guiding the user through five steps: 1. Select user (Nishant Thacker), 2. Select a role (Data Lake Analytics Developer), 3. Select catalog permissions (highlighted in blue), 4. Select file permissions, and 5. Assign selected permissions. To the right, a 'Select catalog permissions' dialog box is displayed, listing 'SCOPE' and 'PERMISSIONS' for the 'ntadlanalytics (Catalog)' and 'master (Database)' objects, both set to 'Read and write'.

# Provisioning users for Data Lake Store

Users must be explicitly granted *read*, *write* or *execute* permission on the Data Lake account.

Note, permissions are automatically set for new co-created Data Lake accounts.

The screenshot shows the Azure portal interface for provisioning a user named Nishant Thacker. It consists of two main windows:

- Add User Wizard (ntadlanalytics - PREVIEW)**: A step-by-step wizard:
  - Select user: Nishant Thacker (Completed)
  - Select a role: Data Lake Analytics Developer (Completed)
  - Select catalog permissions: Selected (Completed)
  - Select file permissions** (Step 4, in progress): This step is highlighted in blue. A red dashed circle surrounds the "APPLY TO" dropdown menu where "This folder and all children" is selected for the "/system" path.  - Assign selected permissions (Step 5, in progress): This step is highlighted in blue.
- Select file permissions (PREVIEW)**: A list of accounts:

NAME
<input checked="" type="checkbox"/> ntadstore

# Creating, submitting a new job

The screenshot shows the Azure Data Lake Analytics portal. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems. A dashed orange circle highlights the '+ New Job' button in the top navigation bar. The main area is titled 'New U-SQL Job' and shows a code editor with the following U-SQL script:

```
1 //Define schema of file, must map all columns
2 @searchlog =
3     EXTRACT UserId      int,
4         Start        DateTime,
5         Region       string,
6         Query        string,
7         Duration    int,
8        Urls         string,
9         ClickedUrls string
10    FROM @"/Samples/Data/SearchLog.tsv"
11    USING Extractors.Tsv();
12
13 OUTPUT @searchlog
14     TO @"/Samples/Output/SearchLog_output.tsv"
15     USING Outputters.Tsv();
```

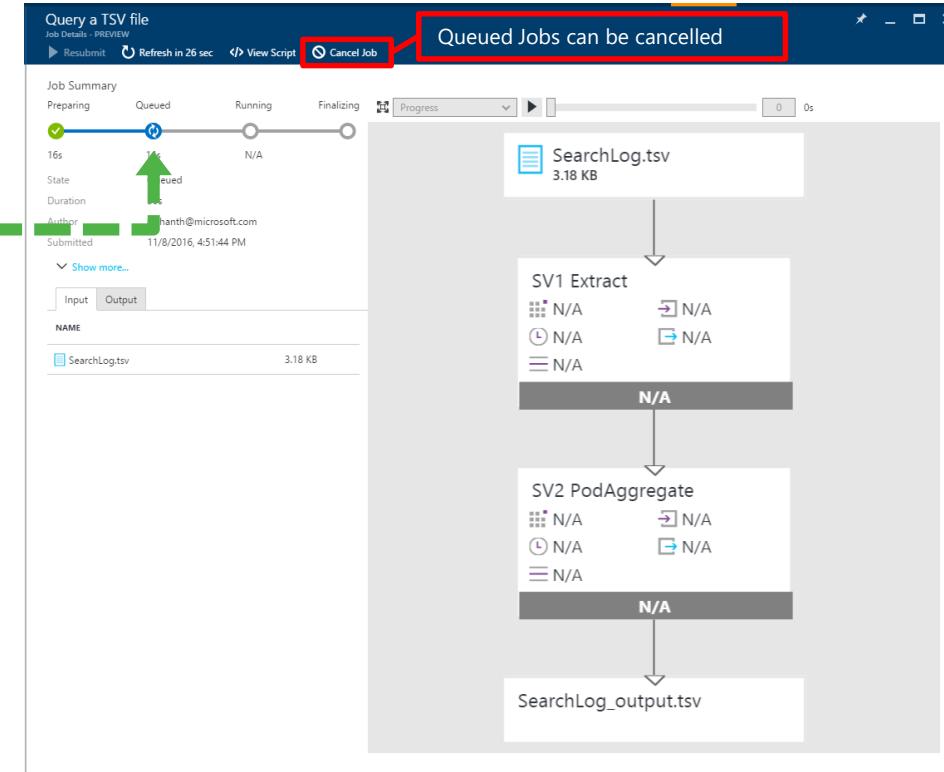
Annotations explain various settings:

- A green arrow points to the 'Priority' input field (set to 1000) with the text: "Priority of the Job in the queue. Lower number means higher priority".
- A green arrow points to the 'Parallelism' slider (set to 20) with the text: "You can specify the max number of compute processes that can execute at a time. Higher degree of parallelism may increase performance, but at a higher cost."

**Job code (U-SQL)  
can be directly  
authored here**

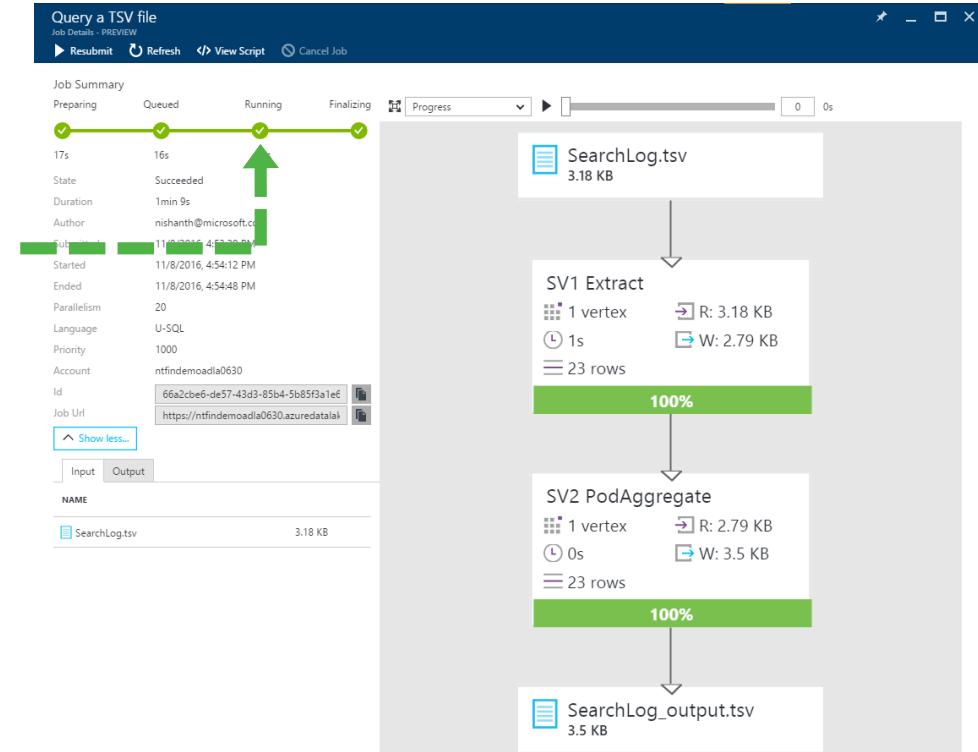
# Job status: queued & running

Initially the job is in "Queued" state, waiting to run



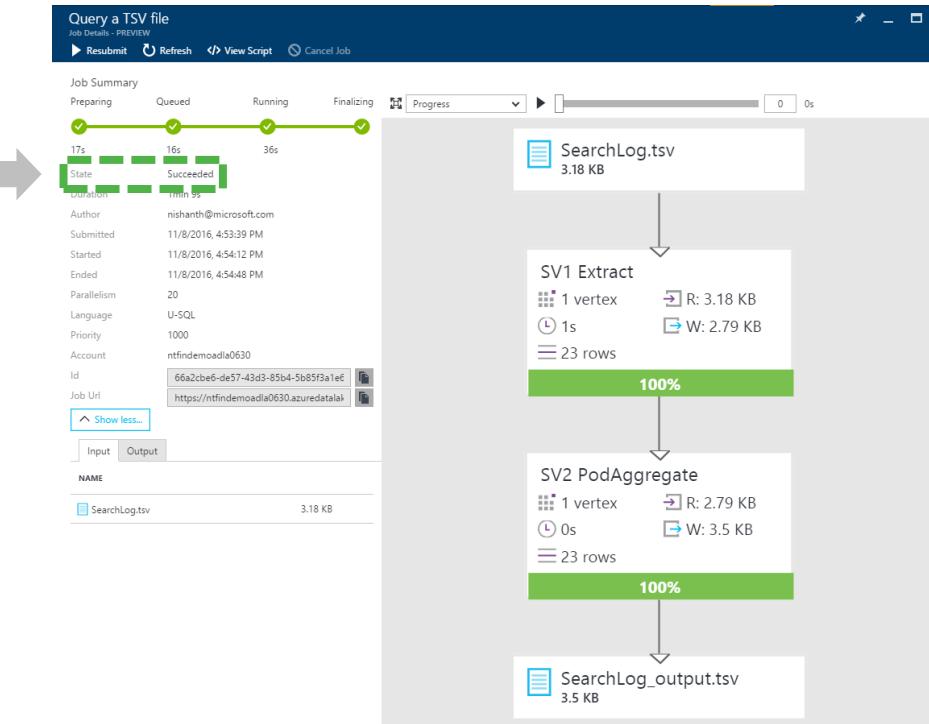
Queued Jobs can be cancelled

After the job is scheduled to run, the job status is "Running"

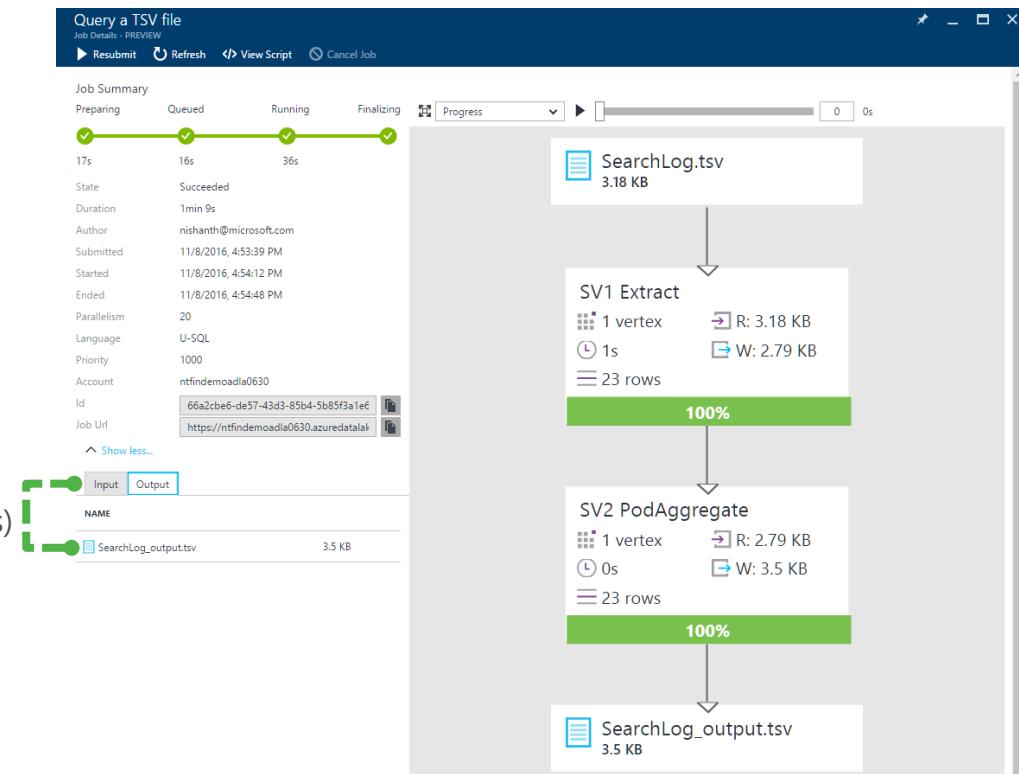


# Job results

If the job completes  
the *final* status is  
succeeded

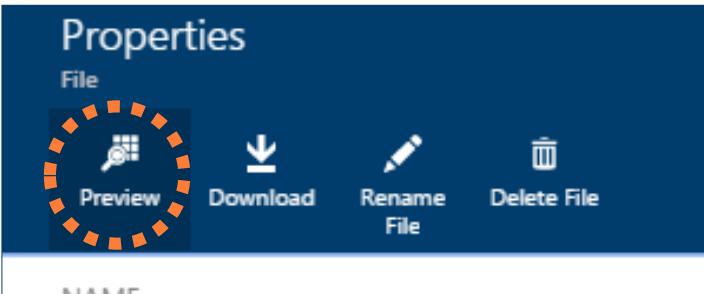


Input and  
output (results)  
are shown



# File preview

- ⚡ Input and output files can be previewed directly in the portal without having to download them
- ⚡ The preview shows the first few rows
- ⚡ Column numbers are automatically assigned
- ⚡ Understands CSV and TSV formats



The screenshot shows a 'Properties' dialog box for a file named 'SearchLog.tsv - PREVIEW'. The dialog has tabs for File, Preview, Download, Rename File, and Delete File. The Preview tab is active, displaying the first few rows of the CSV/TSV file. The columns are labeled 0, 1, 2, 3, 4, 5, and 6. The data rows show various web links and their metadata.

0	1	2	3	4	5	6
399266	2/15/2012 11:53:16 AM	en-us	how to make na...	73	www.nachos.com;www.wikipedia.com	NULL
382045	2/15/2012 11:53:18 AM	en-gb	best ski resorts	614	skiresorts.com;ski-europe.com;www.travelersdigest.com/ski_res...	ski-europe.com;www.travelersdigest.com/ski_res...
382045	2/16/2012 11:53:20 AM	en-gb	broken leg	74	mayoclinic.com/health/webmd.com/a-to-z-guides/mybrokenleg...	mayoclinic.com/health/webmd.com/a-to-z-guides/mybrokenleg...
106479	2/16/2012 11:53:50 AM	en-ca	south park epis...	24	southparkstudios.com;wikipedia.org/wiki/South_Park;imdb.com/ti...	southparkstudios.com
906441	2/16/2012 11:54:01 AM	en-us	cosmos	1213	cosmos.com;wikipedia.org/wiki/Cosmos:_A_Personal_Voyage;hu...	NULL
351530	2/16/2012 11:54:01 AM	en-fr	microsoft	241	microsoft.com;wikipedia.org/wiki/Microsoft;xbox.com	NULL
640806	2/16/2012 11:54:02 AM	en-us	wireless headph...	502	www.amazon.com;reviews.cnet.com/wireless-headphones;store...	www.amazon.com;store...
304305	2/16/2012 11:54:03 AM	en-us	dominos pizza	60	dominos.com;wikipedia.org/wiki/Domino's_Pizza;facebook.com/...	dominos.com
460748	2/16/2012 11:54:04 AM	en-us	yelp	1270	yelp.com;apple.com/us/app/yelp;wikipedia.org/wiki/Yelp,_Inc.;fa...	yelp.com
354841	2/16/2012 11:59:01 AM	en-us	how to run	610	running.about.com;ehow.com/go.com	running.about.com;ehow.com/go.com
354068	2/16/2012 12:00:33 PM	en-mx	what is sql	422	wikipedia.org/wiki/SQL;sqlcourse.com/intro.html;wikipedia.org/...	wikipedia.org/wiki/SQL
674364	2/16/2012 12:00:55 PM	en-us	mexican food re...	283	eltoreador.com;yelp.com/c/redmond-wa/mexican;agavorest.com	NULL
347413	2/16/2012 12:11:55 PM	en-gr	microsoft	305	microsoft.com;wikipedia.org/wiki/Microsoft;xbox.com	NULL
848434	2/16/2012 12:12:35 PM	en-ch	facebook	10	facebook.com;facebook.com/login;wikipedia.org/wiki/Facebook	facebook.com
604846	2/16/2012 12:13:55 PM	en-us	wikipedia	612	wikipedia.org;en.wikipedia.org;en.wikipedia.org/wiki/Wikipedia	wikipedia.org
840614	2/16/2012 12:13:56 PM	en-us	xbox	1220	xbox.com;en.wikipedia.org/wiki/Xbox;xbox.com/xbox360	xbox.com/xbox360
656666	2/16/2012 12:15:55 PM	en-us	hotmail	691	hotmail.com;login.live.com;msn.com;en.wikipedia.org/wiki/Hot...	NULL
951513	2/16/2012 12:17:00 PM	en-us	pokemon	63	pokemon.com;pokemon.com/us;serebii.net	pokemon.com
350350	2/16/2012 12:18:17 PM	en-us	wolfram	30	wolframalpha.com;wolfram.com;mathworld.wolfram.com;en.wiki...	NULL

# ADLA Billing

# ADLA billing

- ⚡ Accounts are **FREE!**
- ⚡ Pay for the compute resources you want for your **queries**
- ⚡ Pay for **storage separately**



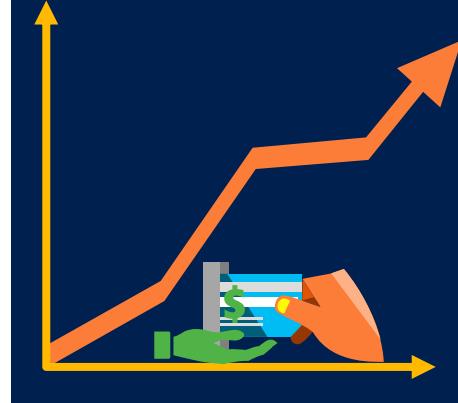
$(\text{query\_hours} * \text{parallelism}) * \text{price/hour}$

USAGE	PREVIEW PRICE (UNTIL DECEMBER 31 <sup>ST</sup> , 2016)	GA PRICE (STARTING JANUARY 1 <sup>ST</sup> , 2017)*
ADLAU	\$1 / hour	\$2 / hour
Completed Job	\$0.025 / Job	Free

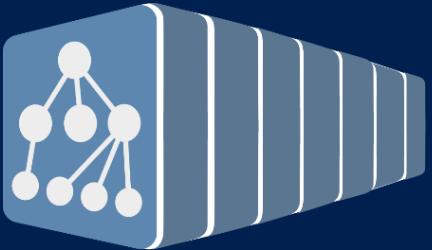
\*special monthly commitment discounted pricing available

# Azure Data Lake Analytics

Start in seconds  
Scale instantly  
Pay per job



Develop massively parallel programs with simplicity



Debug and optimize your Big Data programs with ease



Virtualize your analytics



Enterprise-grade security, auditing and support



# Visual Studio integration

# What can you do with Visual Studio?



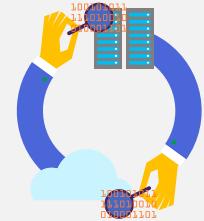
Author U-SQL  
scripts (with  
C# code)



Debug U-SQL and  
C# code



Submit and cancel  
U-SQL Jobs



Visualize physical  
plan of U-SQL  
query



Visualize and  
replay progress  
of job



Fine-tune query  
performance

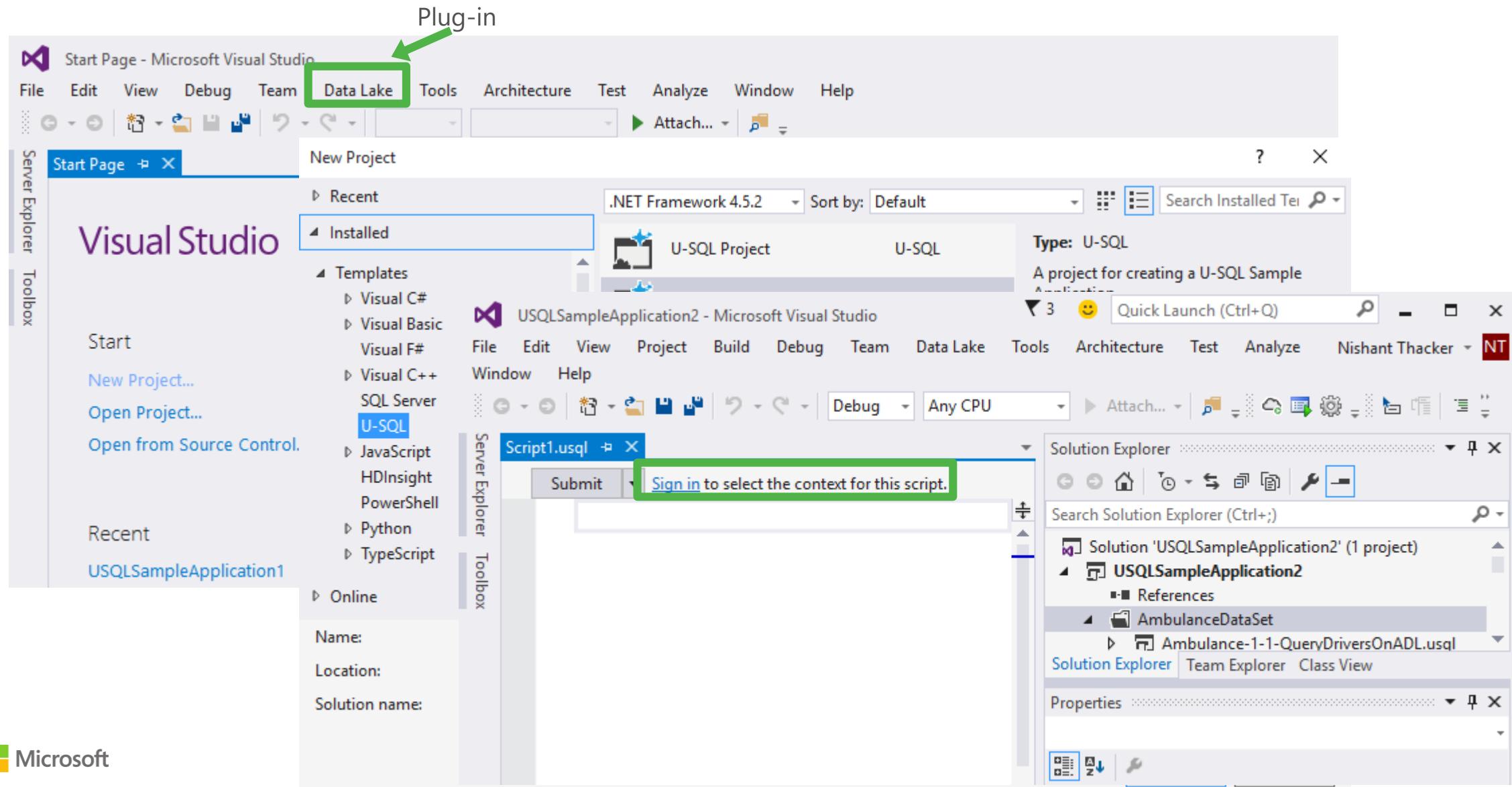


Create metadata  
objects



Browse metadata  
catalog

# How to get going with ADL Tools for Visual Studio



# Authoring U-SQL queries

Visual Studio fully supports authoring U-SQL scripts

While editing, it provides:

- ⚡ IntelliSense
- ⚡ Syntax color coding
- ⚡ Syntax checking
- ⚡ ...

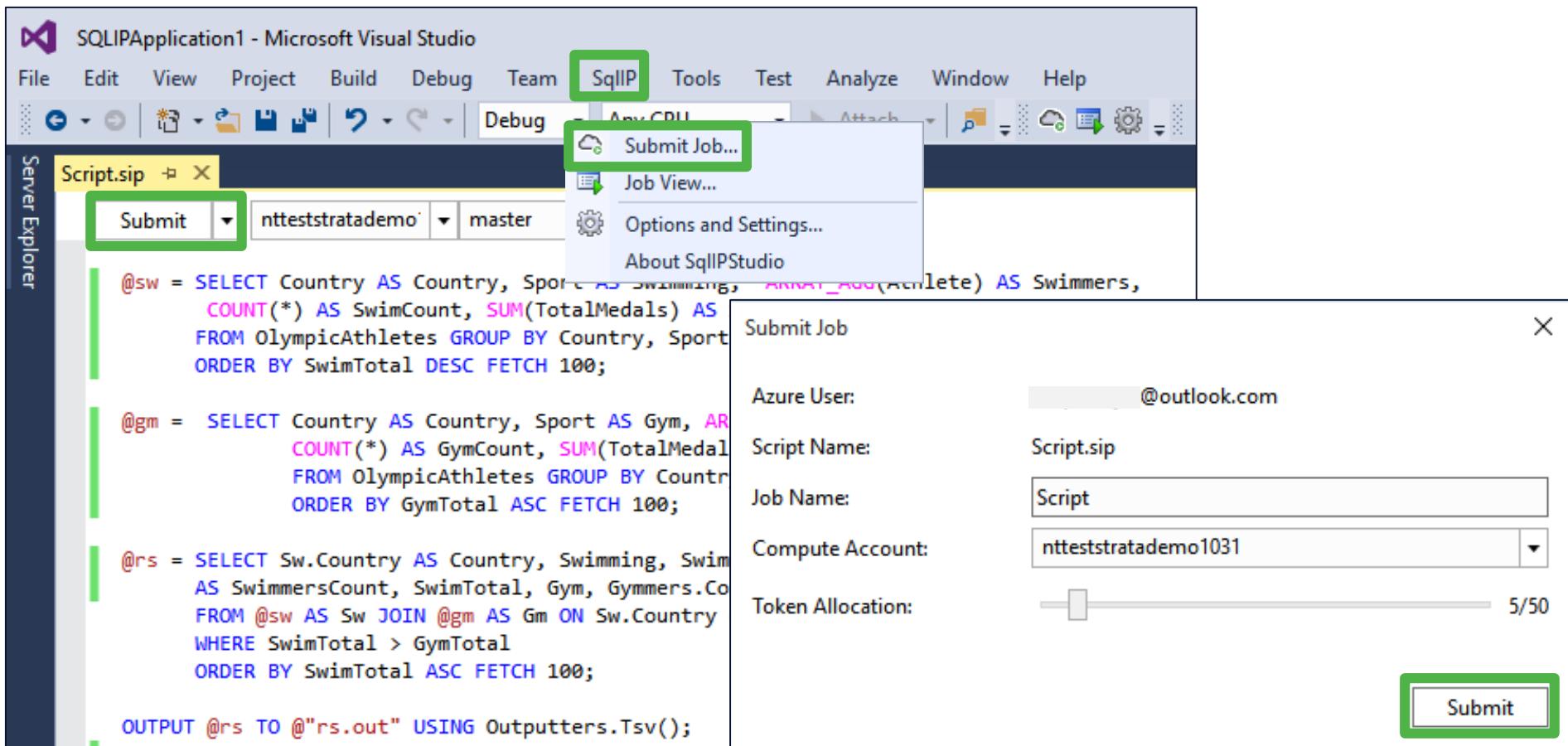
The screenshot shows the Microsoft Visual Studio interface for a U-SQL project named "USQLSampleApplication1". The main window displays a U-SQL script titled "Ambulance-1-1-Qu...DriversOnADL.usql". The script contains several U-SQL statements, including parameter declarations, an EXTRACT clause, and a SELECT query. A contextual menu is open over the word "Csv" in the script, with the label "Contextual menu" highlighted by a red box. The menu options visible include "Csv (+2 overloads)", "Equals", "ReferenceEquals", "Text", and "Tsv". The "Server Explorer" and "Solution Explorer" toolbars are visible on the left and right sides of the IDE respectively.

```
/*
Note:
Please run the scripts in the given order (for example running 1-1 first and then run 1-2, so
There are two ways to run this sample.
1. If you want to run this sample in the Azure Data Lake service,
You can load the samples by going to https://portal.azure.com, and
Then the portal will load the samples to your ADL Store account
2. Also, you can run the scripts locally (Pressing Ctrl + F5) to trigger local run. The success
*/
//0. Initialize some parameters/constants
DECLARE @INPUT_DRIVERS string = "/Samples/Data/AmbulanceData/Drivers.txt";
DECLARE @OUTPUT string = "/Samples/Output/drivers.out";
//1. Extract
@Drivers =
    EXTRACT driver_id int,
    name string,
    street string,
    city string,
    region string,
    zipcode string,
    country string,
    phone_numbers string
    FROM @INPUT_DRIVERS
    USING Extractors.Text(delimiter : '\t', quoting : true, encoding : Encoding.Unicode);
//2. Select
@Result =
    SELECT country,
    city,
    COUNT( * ) Csv
    FROM @Drivers
    GROUP BY country,
    city
*/
```

# Submitting a U-SQL job

Jobs can be submitted directly from Visual Studio in two ways

You have to be logged into Azure and have to specify the target Azure Data Lake account.



# Query design

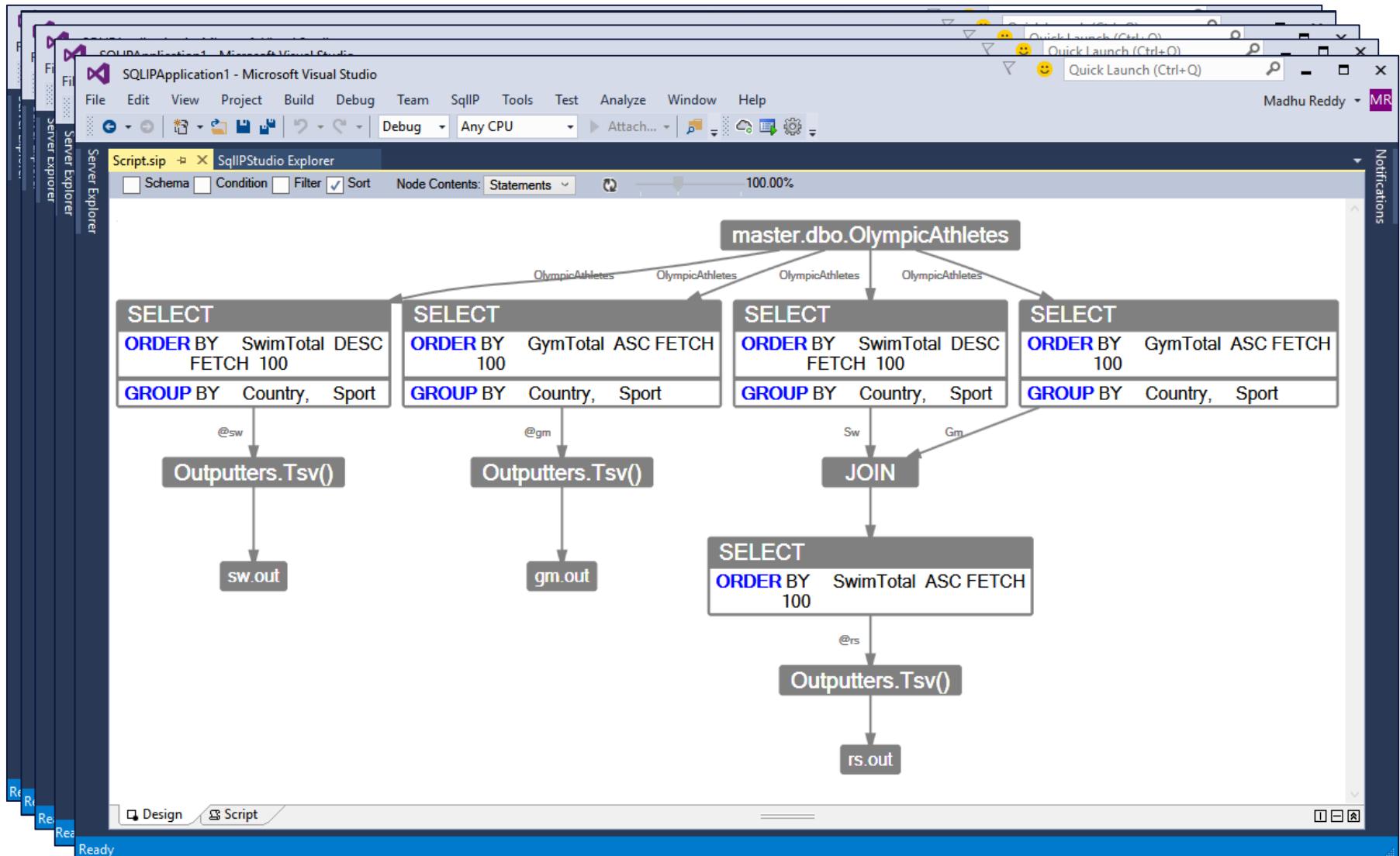
⚡ U-SQL Studio lets you see the logical query design including:

**Schema**

**Join conditions**

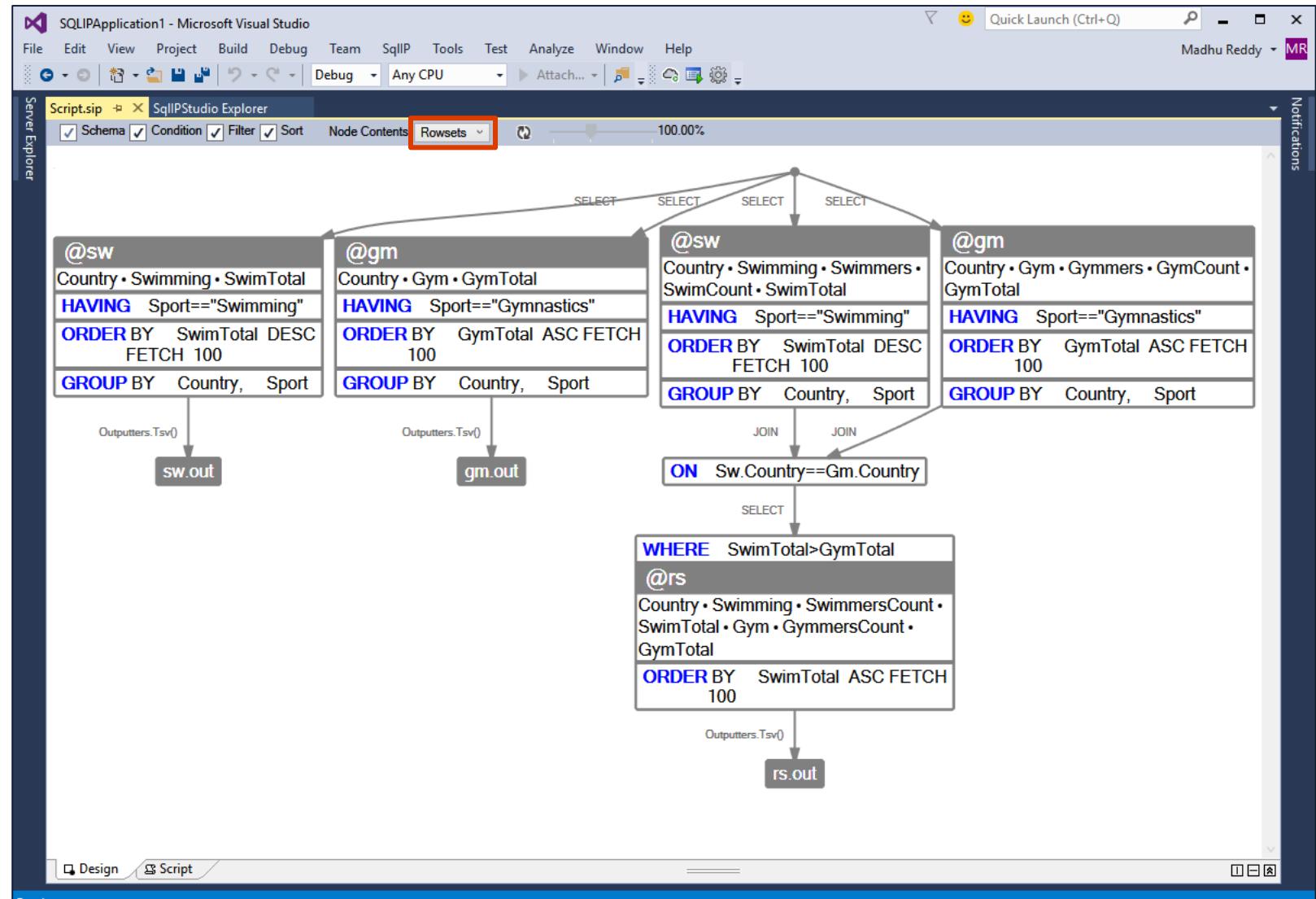
**Filter plan**

**Sort plan**



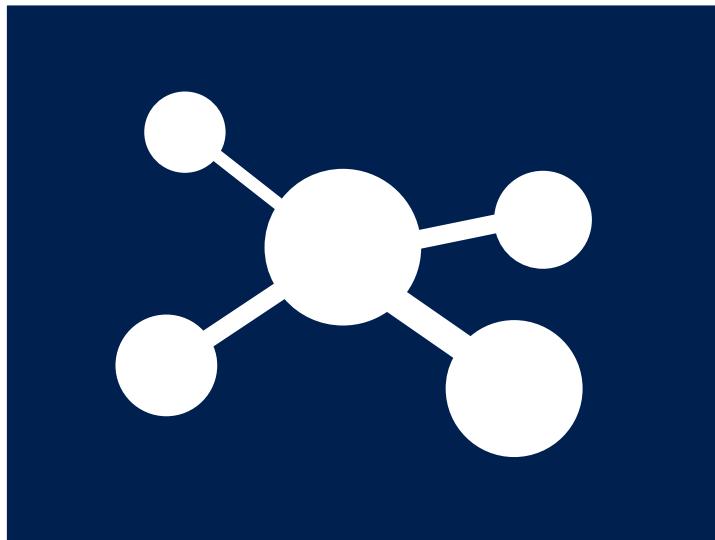
# Query design -RowSet

- The query design can also be visualized in terms of the RowSets and the transformation applied to them.



# Metadata objects

- ⚡ ADL Analytics creates and stores a set of metadata objects in a catalog maintained by a metadata service
- ⚡ Tables and TVFs are created by DDL statements (CREATE TABLE ...)
- ⚡ Metadata objects can be created directly through the Server Explorer



Azure Data Lake Analytics account

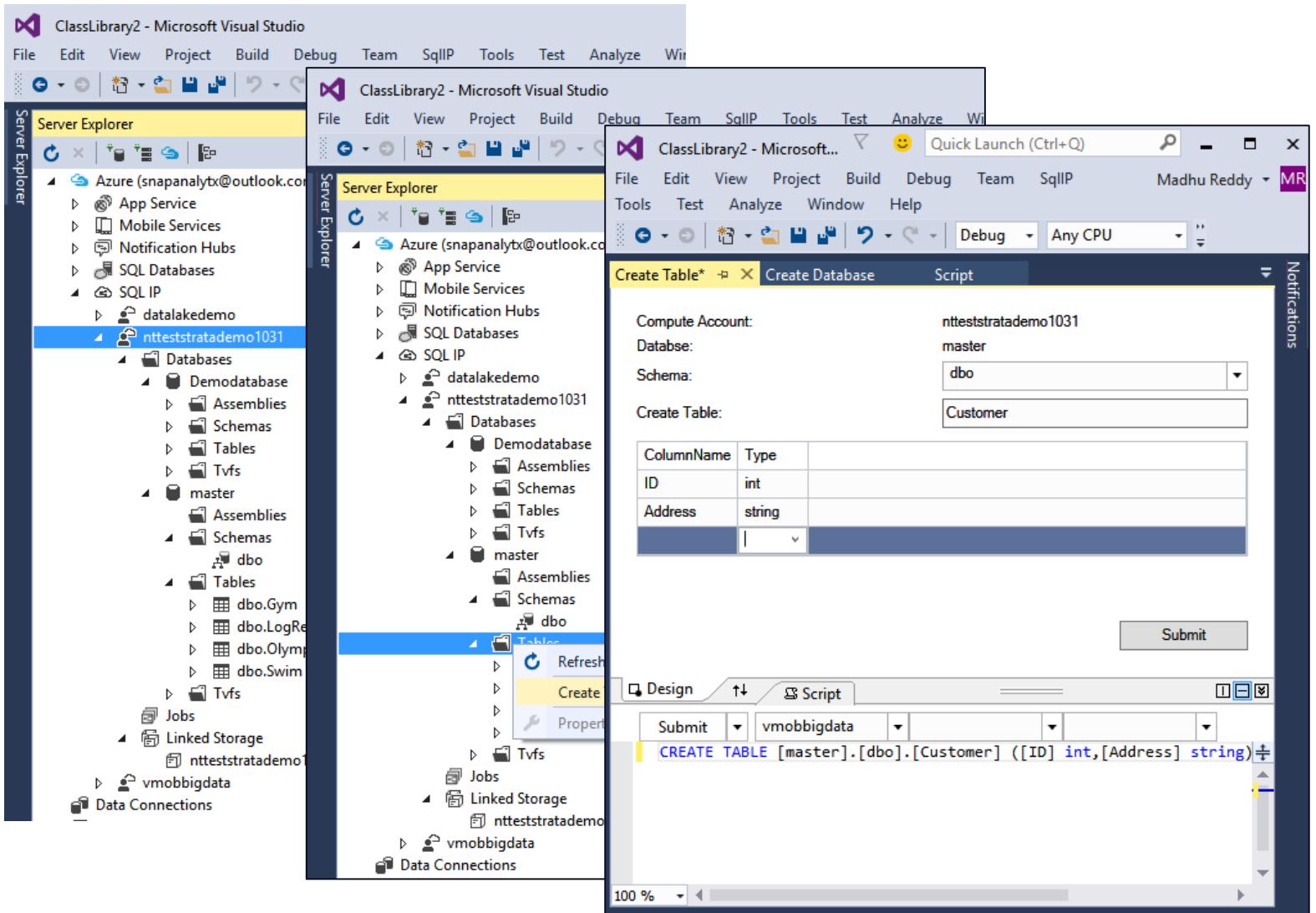
- ⚡ Databases
  - Tables
  - Table valued functions
  - Jobs
  - Schemas
- ⚡ Linked storage

# Metadata catalog

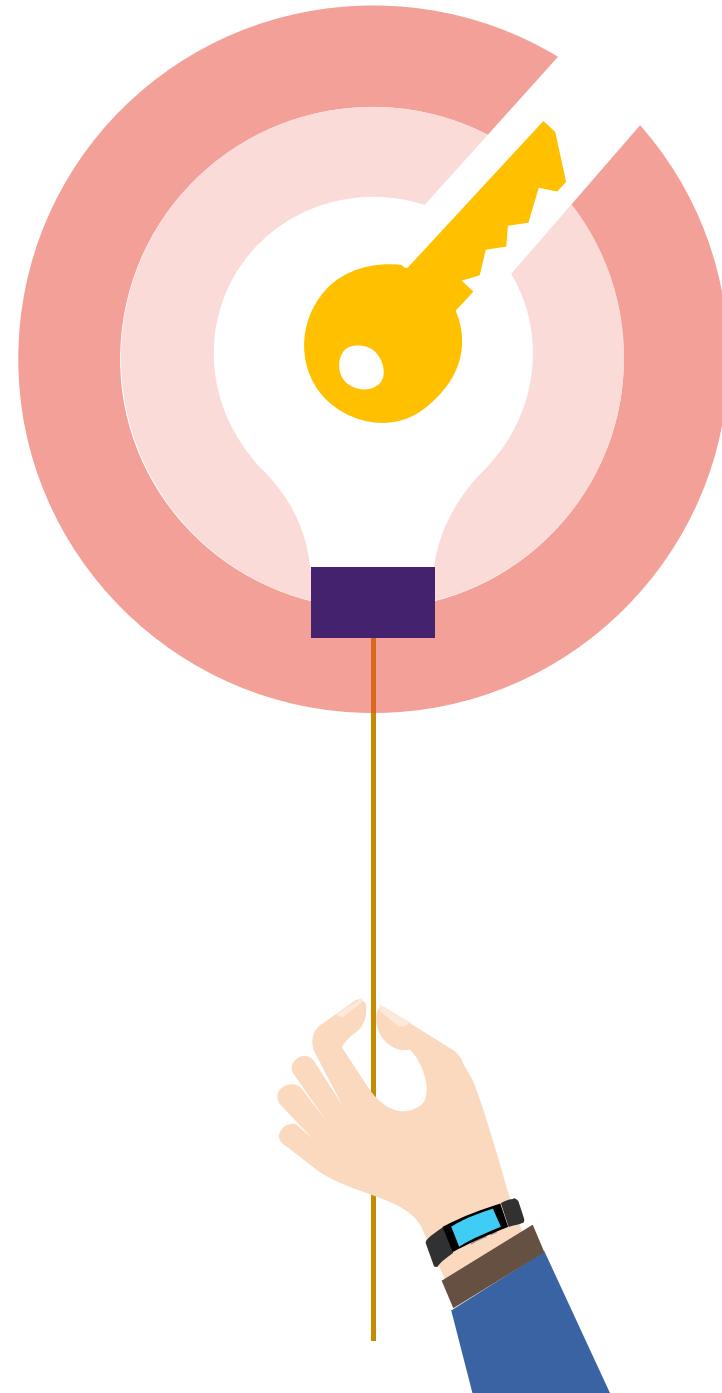
The metadata catalog can be browsed with the Visual Studio Server Explorer

## Server Explorer lets you:

1. Create new tables, schemas and databases
2. Register assemblies

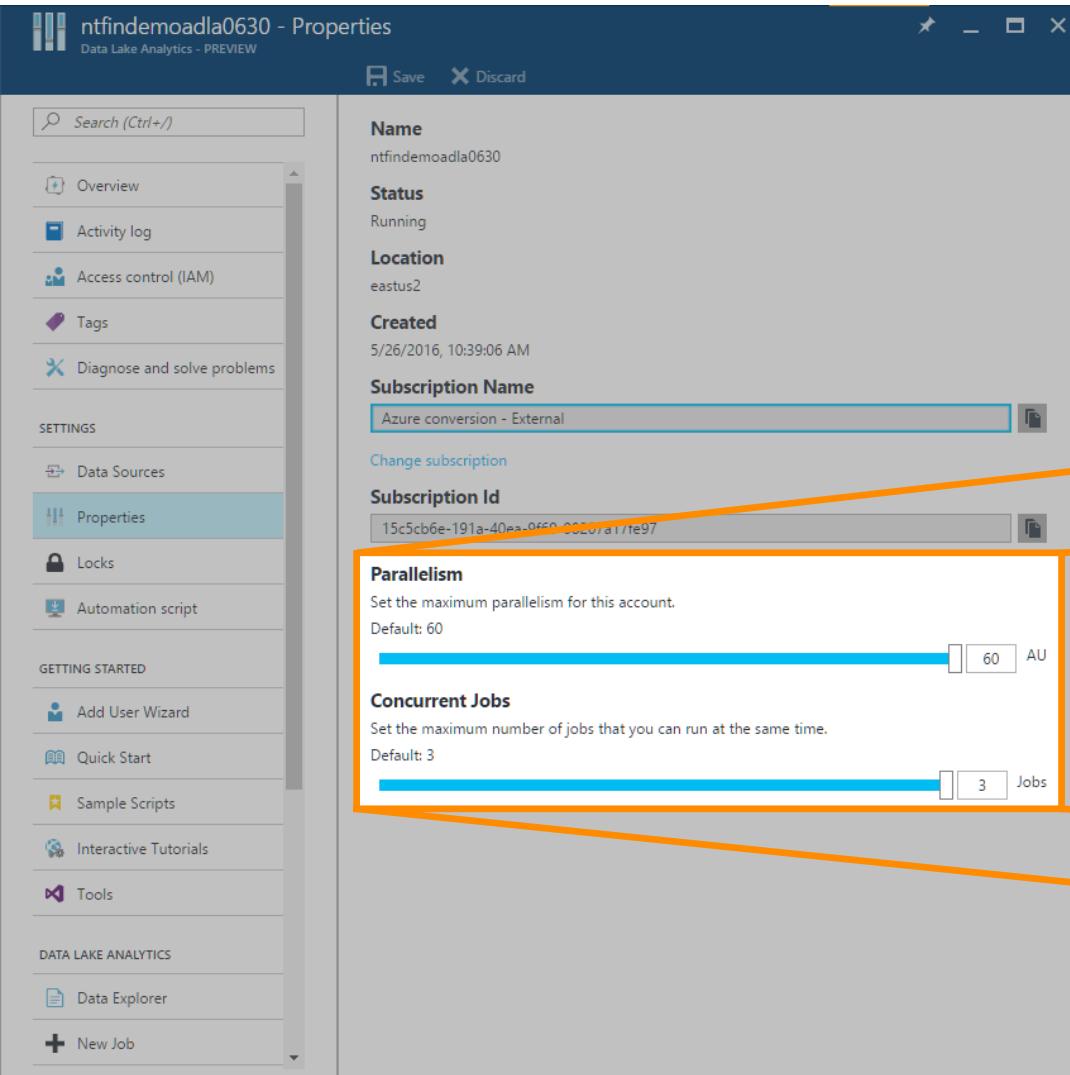


# Key concepts



# Analytics Account Properties

## Azure Data Lake Analytics Unit



Parallelism of  $N = N$  ADLAUs

1 ADLAU  $\approx$  a Container, 2 cores and 6 GB of memory

This is the number of ADLAUs for the account.

### Parallelism

Set the maximum parallelism for this account.

Default: 60

60 AU

### Concurrent Jobs

Set the maximum number of jobs that you can run at the same time.

Default: 3

3 Jobs

IGNORE THIS

# Data Lake account configuration

## ADL Analytics account

Links to ADL Stores

Links to Azure Blob Stores

U-SQL Catalog

Metadata

Job Queue

Query Store



## ADL Store account (the default one)

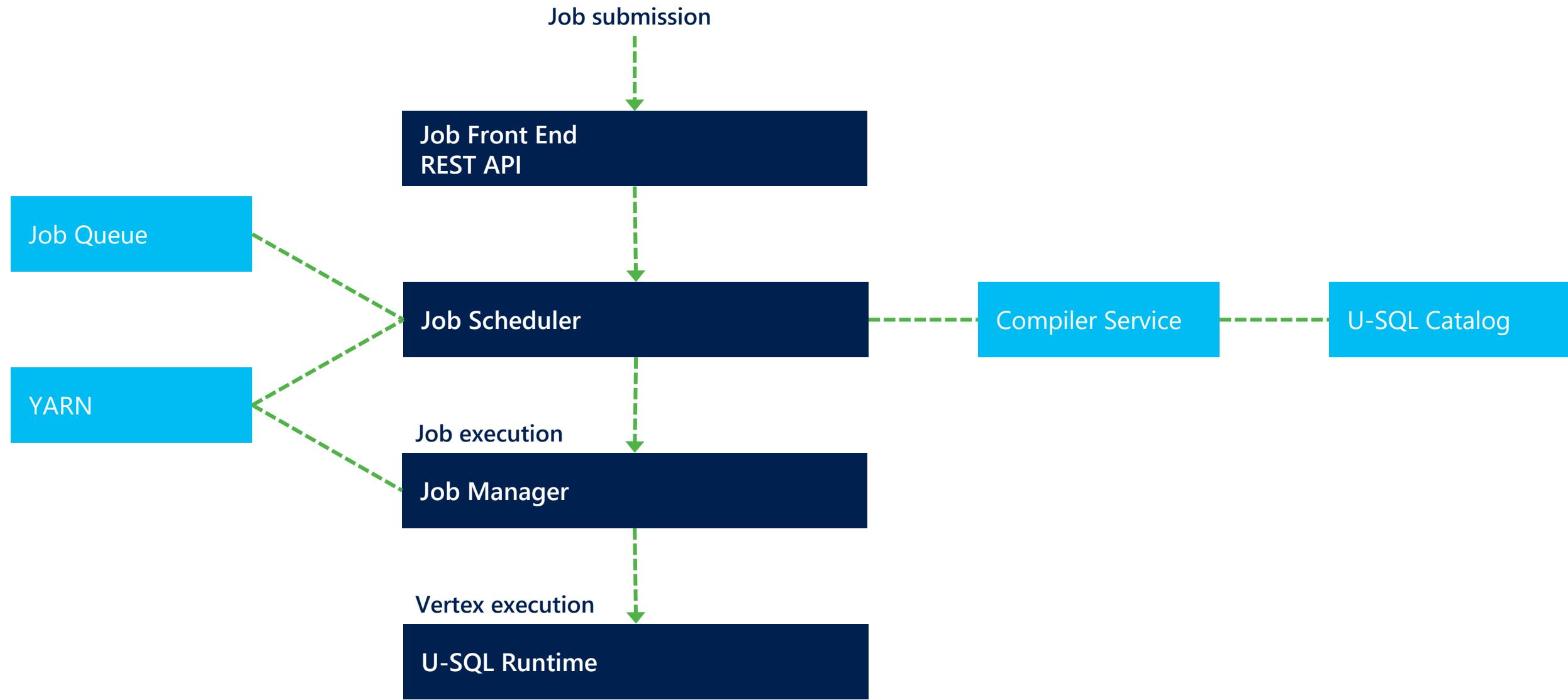
U-SQL Catalog

Data



An ADL Store IS REQUIRED for ADL Analytics to function.

# Simplified job workflow



ntfindemo - Microsoft Visual Studio

File Edit View Project Build Debug Team Data Lake Tools Architecture Test Analyze Window Help

Quick Launch (Ctrl+Q)

Analytics Demo AD

Server Explorer Toolbox Cloud Explorer

Job View: ntfinde... 8deda3ebc95fc4b GenData.usql Job View: ntfinde... 9658-15a1e25db3ed

Job Name: LoadAndProcessData | Load Profile

**Job Summary**

- Preparing
- Queued
- Running
- Finalizing

35 seconds 0 seconds 7.3 hours

**Job Completed Successfully**  
No errors to report.

**Job Result** Succeeded  
**Total Duration** 7.3 hours  
**Submit Time** 11/7/2016 9:45:27 AM  
**Start Time** 11/7/2016 9:46:20 AM  
**End Time** 11/7/2016 5:03:00 PM  
**Compilation** 35 seconds  
**Queued** 0 seconds  
**Running** 7.3 hours  
**Account** ntfindemo\da16630  
**Author** analyticsdemo@outlook.com  
**Priority** 1000  
**Parallelism** 60  
**Bytes Left** 759,092,887,081  
**Bytes Read** 4,390,223,643,208  
**Bytes Written** 7,340,221,434,900  
**Vertices** 1,953

**How does the Parallelism number relate to Vertices?**

**What does Vertices mean?**

**What is this?**

Job Graph

Display: Progress | Succeeded | Failed | Running | Waiting

Job Playback

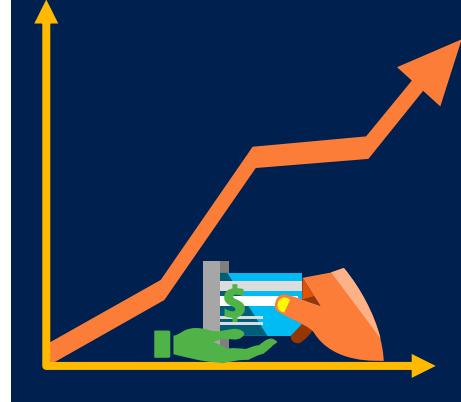
00:00:00

Error List Exception Settings Output

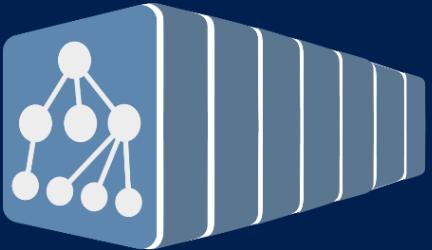
Ready

# Azure Data Lake Analytics

Start in seconds  
Scale instantly  
Pay per job



Develop massively parallel programs with simplicity



Debug and optimize your Big Data programs with ease



Virtualize your analytics



Enterprise-grade security, auditing and support



# Concepts: jobs, stages and vertices

- ⚡ Each job is broken into 'n' number of **vertices**
- ⚡ Each vertex is some work that needs to be done

```
@sw = SELECT Country AS Country, Sport AS Swimming, SUM(TotalMedals) AS SwimTotal
FROM OlympicAthletes GROUP BY Country, Sport HAVING Sport == "Swimming"
ORDER BY SwimTotal DESC FETCH 100;

@gm = SELECT Country AS Country, Sport AS Gym, SUM(TotalMedals) AS GymTotal
FROM OlympicAthletes GROUP BY Country, Sport HAVING Sport == "Gymnastics"
ORDER BY GymTotal ASC FETCH 100;

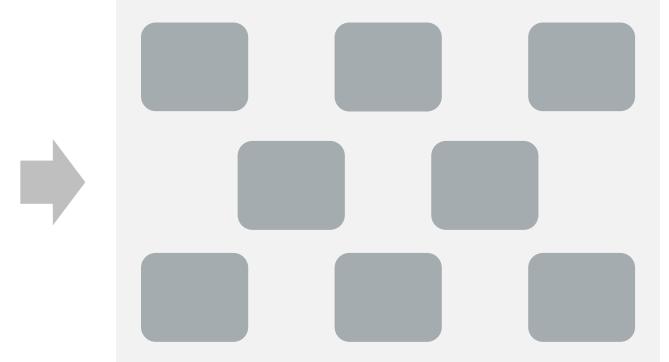
OUTPUT @sw TO @"sw.out" USING Outputters.Tsv();
OUTPUT @gm TO @"gm.out" USING Outputters.Tsv();

@sw = SELECT Country AS Country, Sport AS Swimming, ARRAY_AGG(Athlete) AS Swimmers,
COUNT(*) AS SwimCount, SUM(TotalMedals) AS Swimtotal
FROM OlympicAthletes GROUP BY Country, Sport HAVING Sport == "Swimming"
ORDER BY SwimTotal DESC FETCH 100;

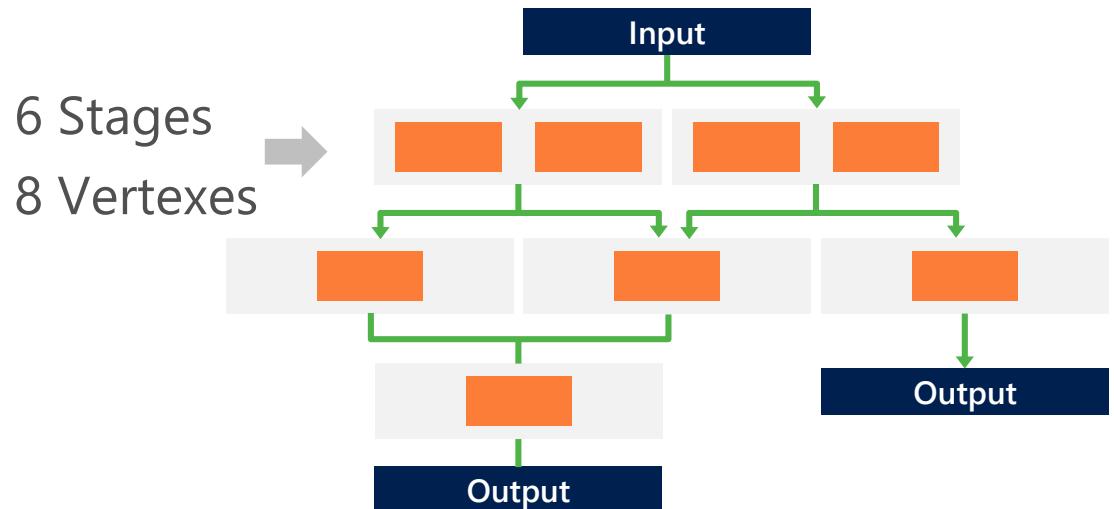
@gm = SELECT Country AS Country, Sport AS Gym, ARRAY_AGG(Athlete) AS Gymmers,
COUNT(*) AS GymCount, SUM(TotalMedals) AS GymTotal
FROM OlympicAthletes GROUP BY Country, Sport HAVING Sport == "Gymnastics"
ORDER BY GymTotal ASC FETCH 100;

@rs = SELECT Sw.Country AS Country, Swimming, Swimmers.Count
AS SwimmersCount, SwimTotal, Gym, Gymmers.Count AS GymmersCount, GymTotal
FROM @sw AS Sw JOIN @gm AS Gm ON Sw.Country == Gm.Country
WHERE SwimTotal > GymTotal
ORDER BY SwimTotal ASC FETCH 100;

OUTPUT @rs TO @"rs.out" USING Outputters.Tsv();
```

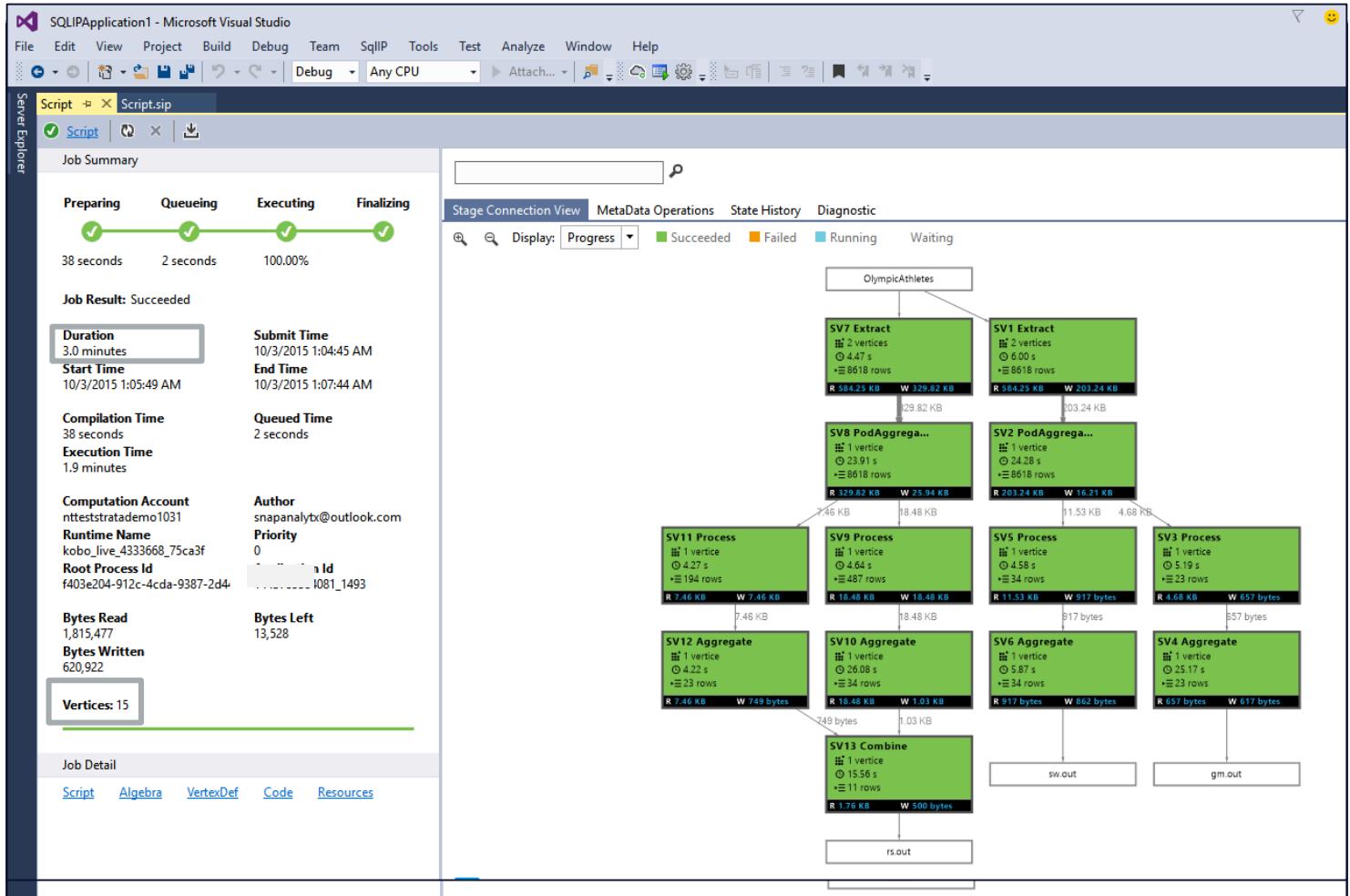


- ⚡ Vertices are organized into **stages**
  - Vertices in each stage do the same work on the same data
  - Vertex in one stage may depend on a vertex in an earlier stage
- ⚡ Stages themselves are organized into an acyclic graph



# Job execution graph

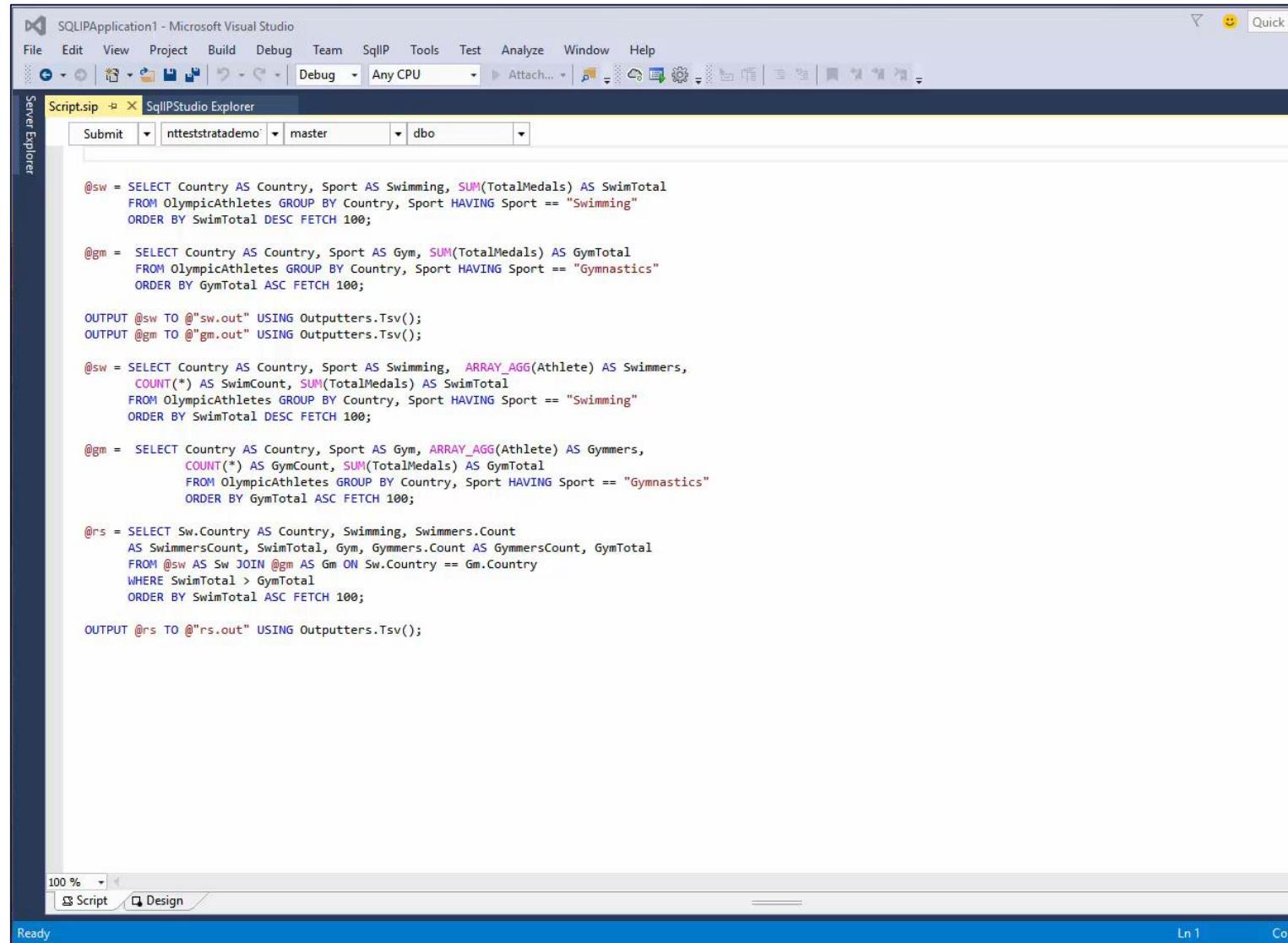
- After a job is submitted the progress of the execution of the job as it goes through the different stages is shown and updated continuously



- Important stats about the job are also displayed and updated continuously



# Job execution graph (video)



The screenshot shows a Microsoft Visual Studio interface with the title bar "SQLApplication1 - Microsoft Visual Studio". The menu bar includes File, Edit, View, Project, Build, Debug, Team, SqlIP, Tools, Test, Analyze, Window, and Help. The toolbar has various icons for file operations like Open, Save, and Print. The status bar at the bottom shows "Ready", "Ln 1", and "Col".

The main window displays a T-SQL script named "Script.sip" under the "Server Explorer" tab. The script is as follows:

```
@sw = SELECT Country AS Country, Sport AS Swimming, SUM(TotalMedals) AS SwimTotal
FROM OlympicAthletes GROUP BY Country, Sport HAVING Sport == "Swimming"
ORDER BY SwimTotal DESC FETCH 100;

@gm = SELECT Country AS Country, Sport AS Gym, SUM(TotalMedals) AS GymTotal
FROM OlympicAthletes GROUP BY Country, Sport HAVING Sport == "Gymnastics"
ORDER BY GymTotal ASC FETCH 100;

OUTPUT @sw TO @"sw.out" USING Outputters.Tsv();
OUTPUT @gm TO @"gm.out" USING Outputters.Tsv();

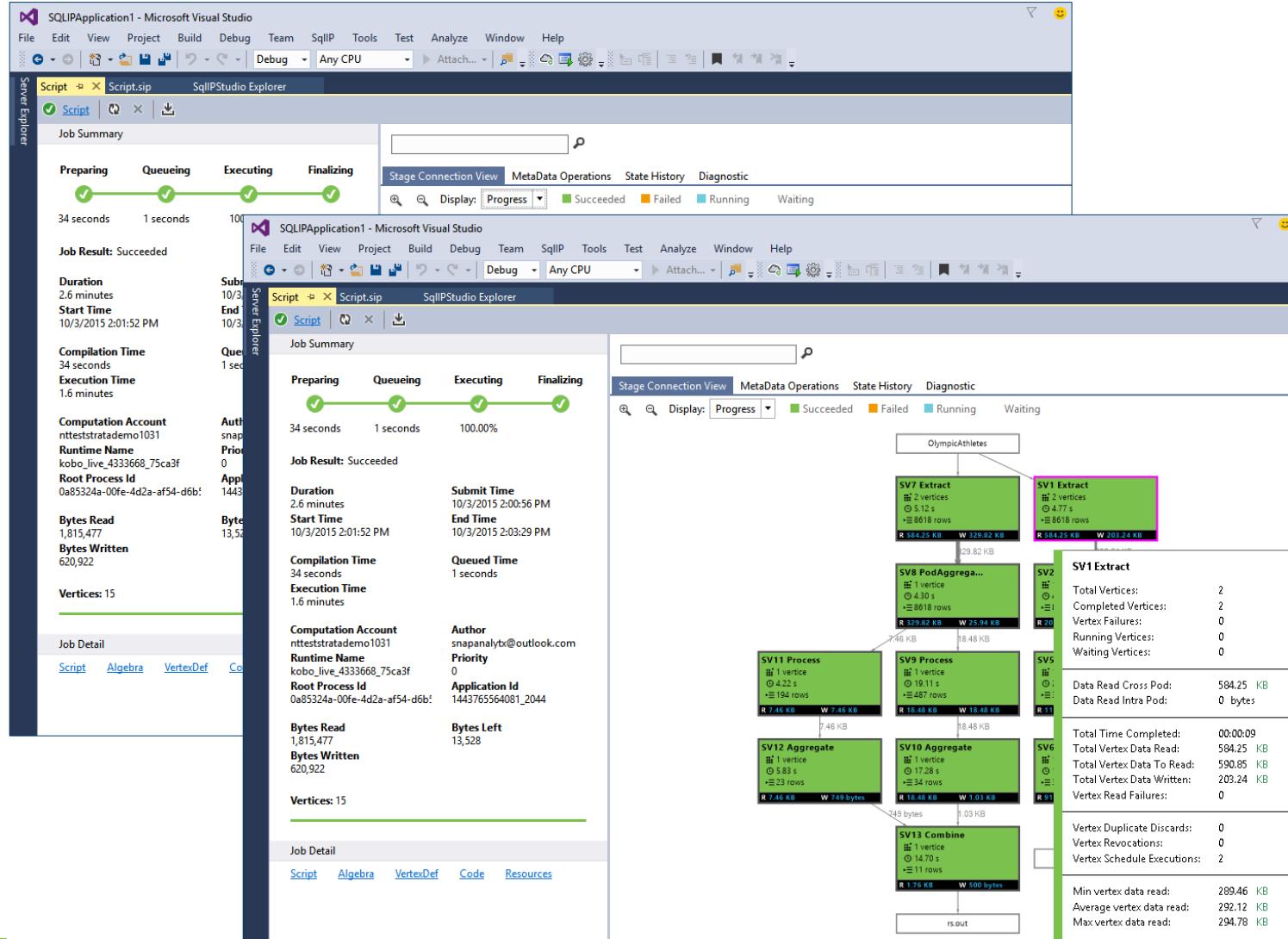
@sw = SELECT Country AS Country, Sport AS Swimming, ARRAY_AGG(Athlete) AS Swimmers,
COUNT(*) AS SwimCount, SUM(TotalMedals) AS SwimTotal
FROM OlympicAthletes GROUP BY Country, Sport HAVING Sport == "Swimming"
ORDER BY SwimTotal DESC FETCH 100;

@gm = SELECT Country AS Country, Sport AS Gym, ARRAY_AGG(Athlete) AS Gymmers,
COUNT(*) AS GymCount, SUM(TotalMedals) AS GymTotal
FROM OlympicAthletes GROUP BY Country, Sport HAVING Sport == "Gymnastics"
ORDER BY GymTotal ASC FETCH 100;

@rs = SELECT Sw.Country AS Country, Swimming, Swimmers.Count
AS SwimmersCount, SwimTotal, Gym, Gymmers.Count AS GymmersCount, GymTotal
FROM @sw AS Sw JOIN @gm AS Gm ON Sw.Country == Gm.Country
WHERE SwimTotal > GymTotal
ORDER BY SwimTotal ASC FETCH 100;

OUTPUT @rs TO @"rs.out" USING Outputters.Tsv();
```

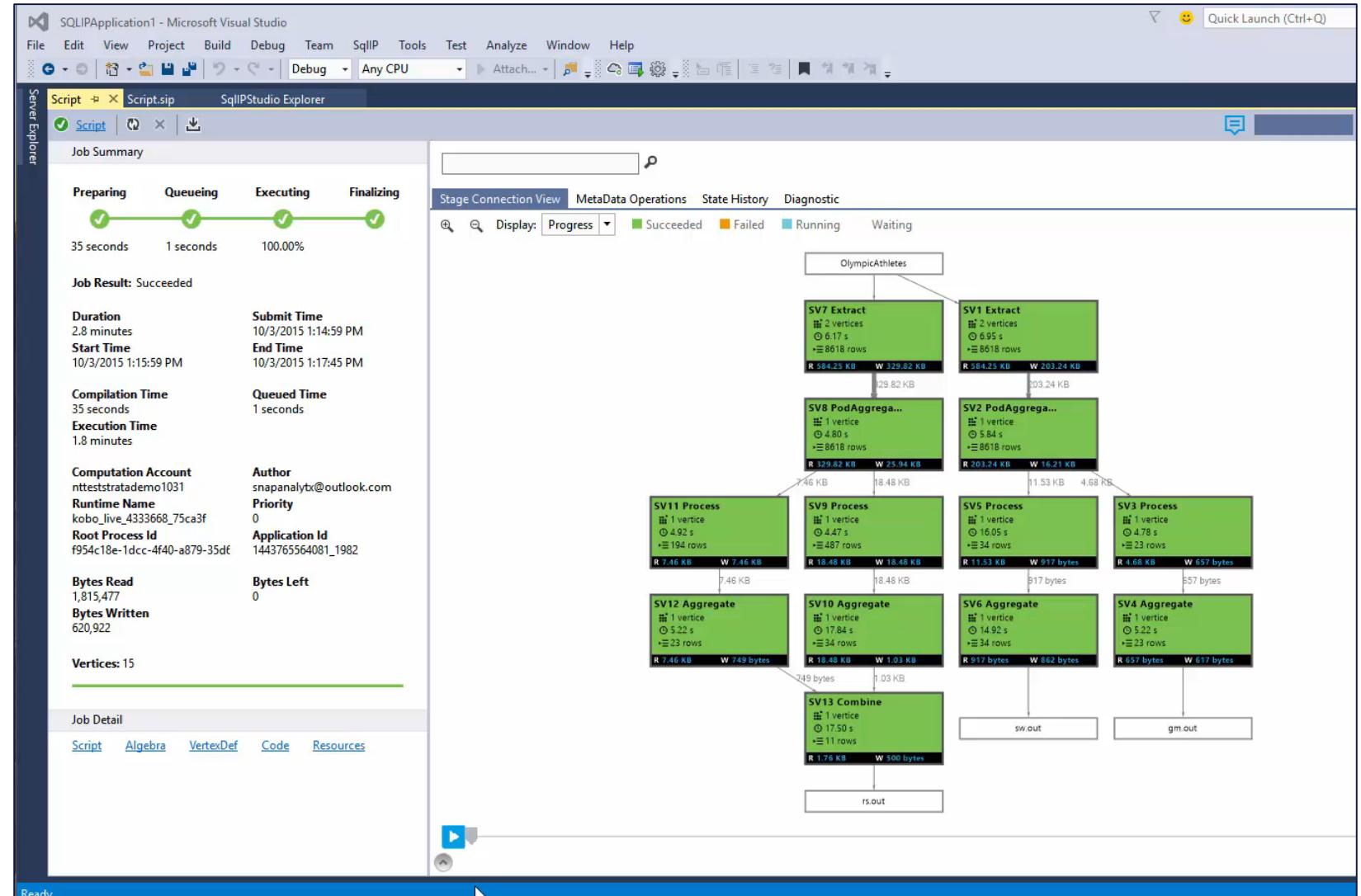
# Job execution graph – node details



Hovering over them, you can get details about the nodes.

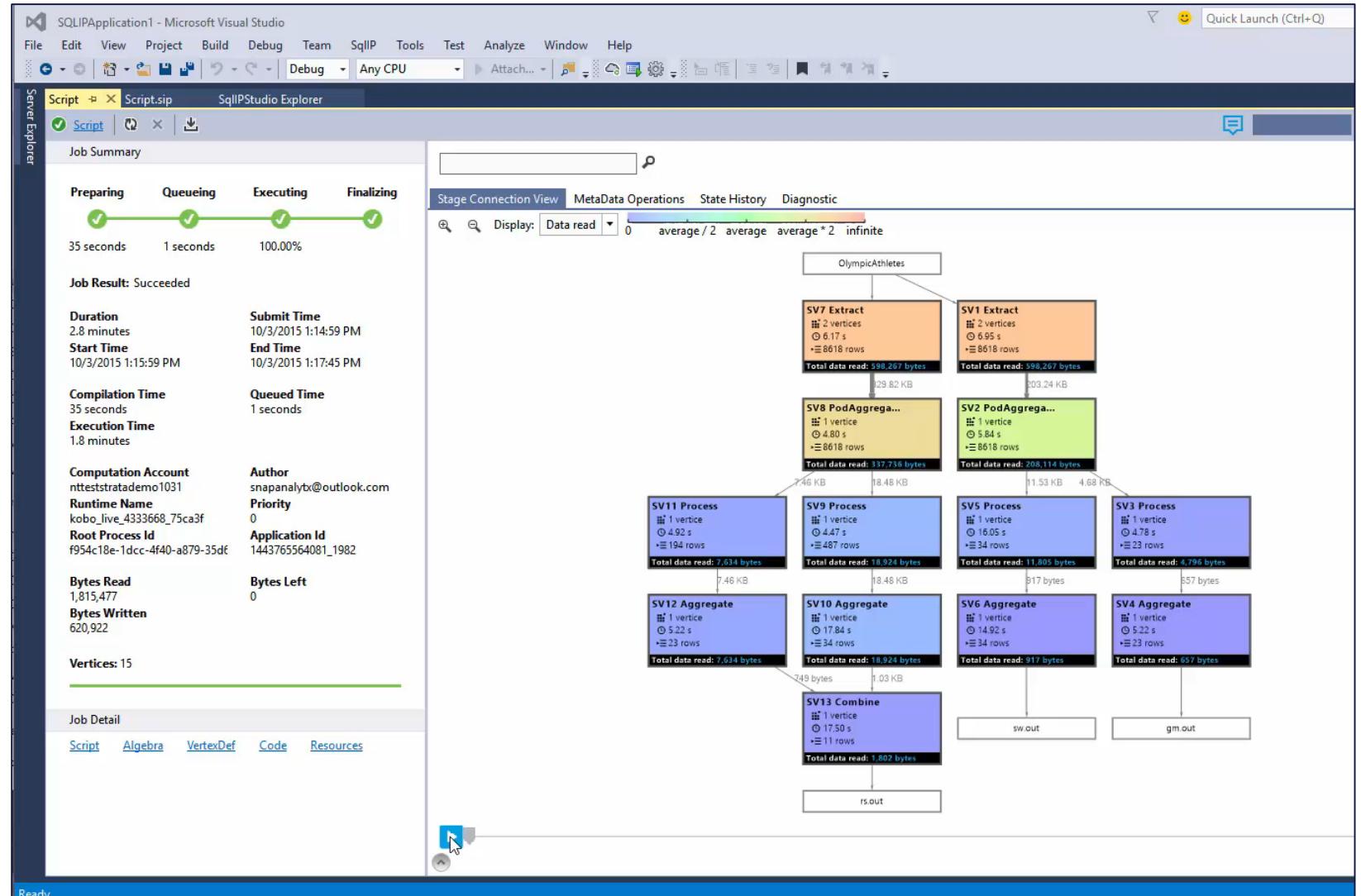
# Job execution “progress” playback (video)

- ⚡ For performance tuning, identify bottlenecks and debugging, you can playback the job execution graph



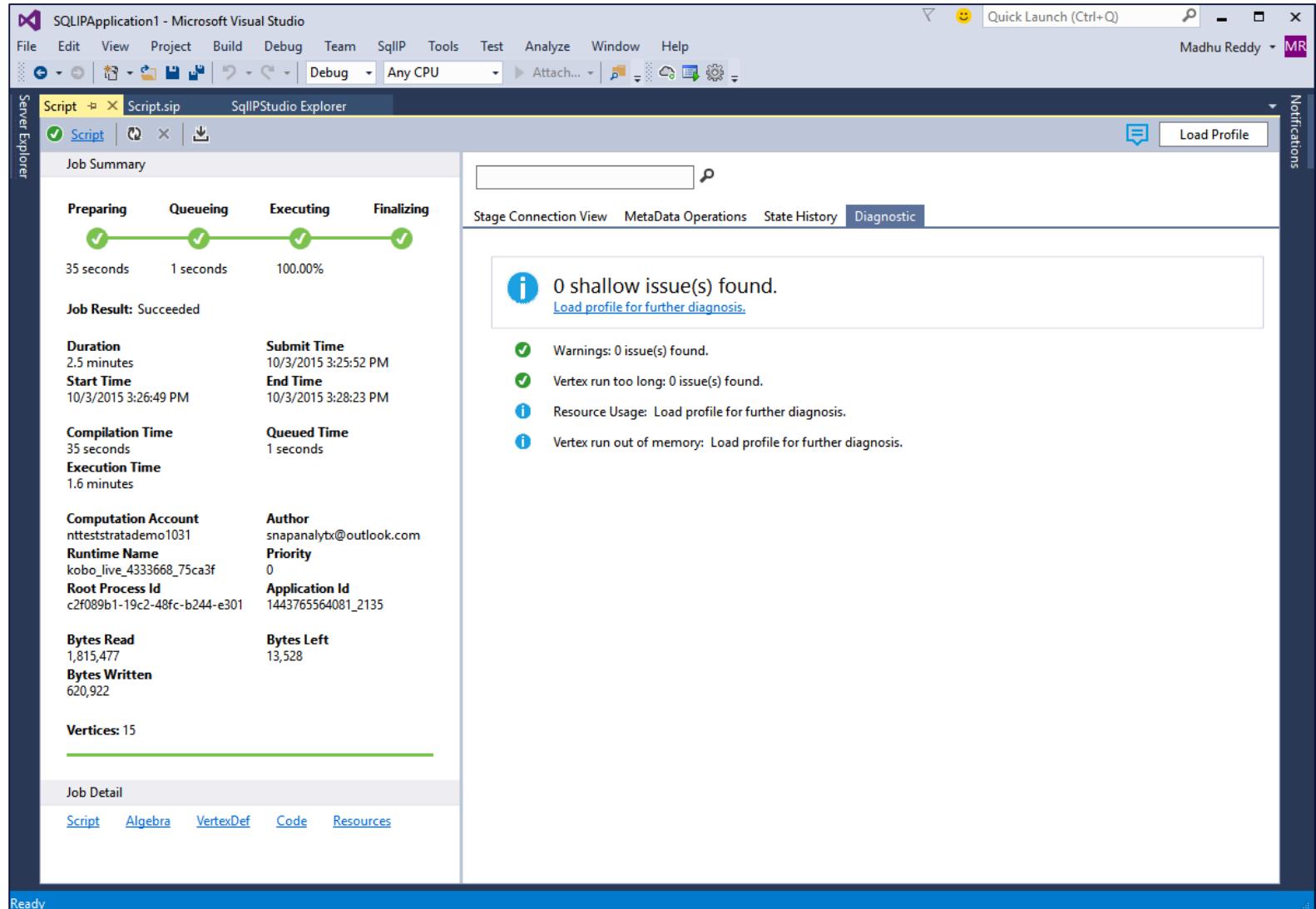
# “Data read” playback (video)

- ⚡ For performance tuning, identify bottlenecks and debugging, you can playback the job execution graph



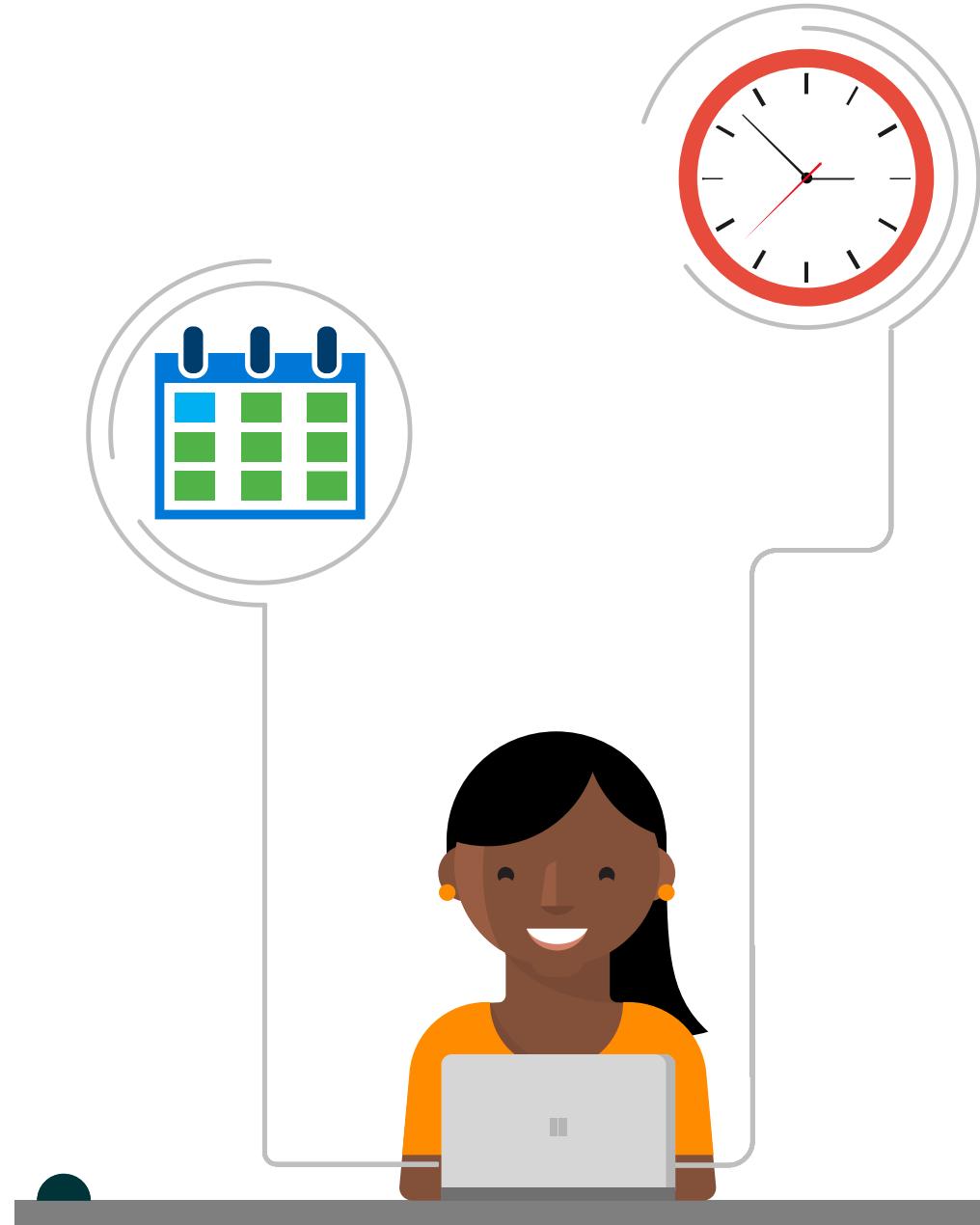
# Job diagnostics

⚡ Diagnostics information is shown to help with debugging and performance issues



# Job scheduling

## States, queue, priority



# Visual Studio: Job states

UX

## Job State

### Preparing

New

Compiling

The script is being compiled by the Compiler Service

### Queued

Queued

All jobs enter the queue.

Scheduling

Are there enough ADLAUs to start the job?

Starting

If yes, then allocate those ADLAUs for the job

### Running

Running

The U-SQL runtime is now executing the code on 1 or more ADLAUs or finalizing the outputs

### Finalizing (Succeeded, Failed, Cancelled)

Ended

The job has concluded.

Job Name: LoadAndProcessData |

### Job Summary

Preparing      Queued      Running      Finalizing

35 seconds      0 seconds      7.3 hours

Job Completed Successfully

No errors to report.

Job Result      Succeeded

Total Duration      7.3 hours

Submit Time      11/7/2016 9:45:27 AM

Start Time      11/7/2016 9:46:20 AM

End Time      11/7/2016 5:03:00 PM

Compilation      35 seconds

Queued      0 seconds

Running      7.3 hours

Account      ntfindemoadla0630

Author      analyticsdemo@outlook.com

Priority      1000

Parallelism      60

Bytes Left      0

Bytes Read      4,390,223,643,208

Bytes Written      7,340,221,434,900

Vertices      1,953

### Job Detail

[Script](#)    [Resources](#)    [Vertex Execution View](#)

# Why does a job get queued?

## Local cause

Possible condition:

- Not enough containers available to your account



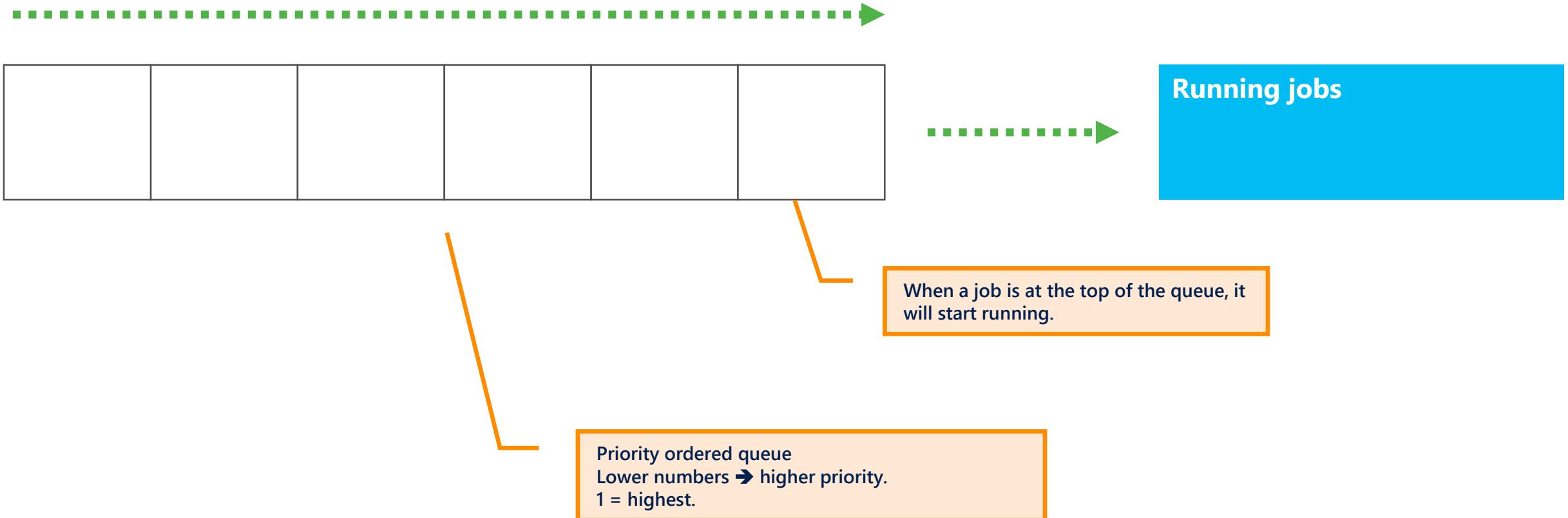
## Global cause (very rare)

Possible conditions:

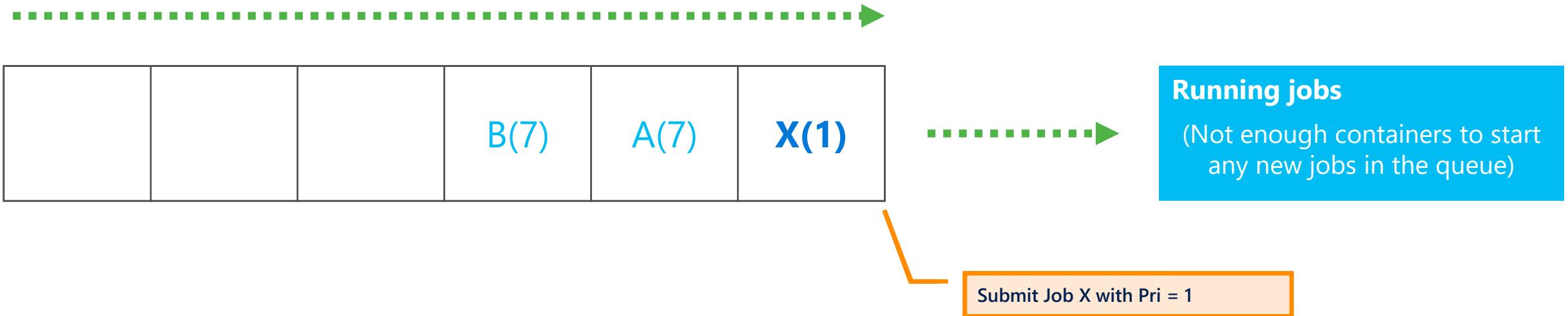
- System-wide shortage of containers
- System-wide shortage of bandwidth



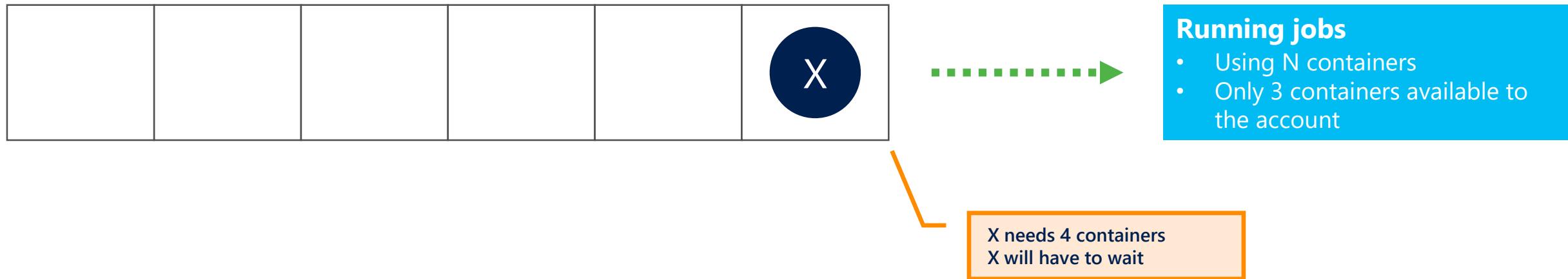
# The job queue



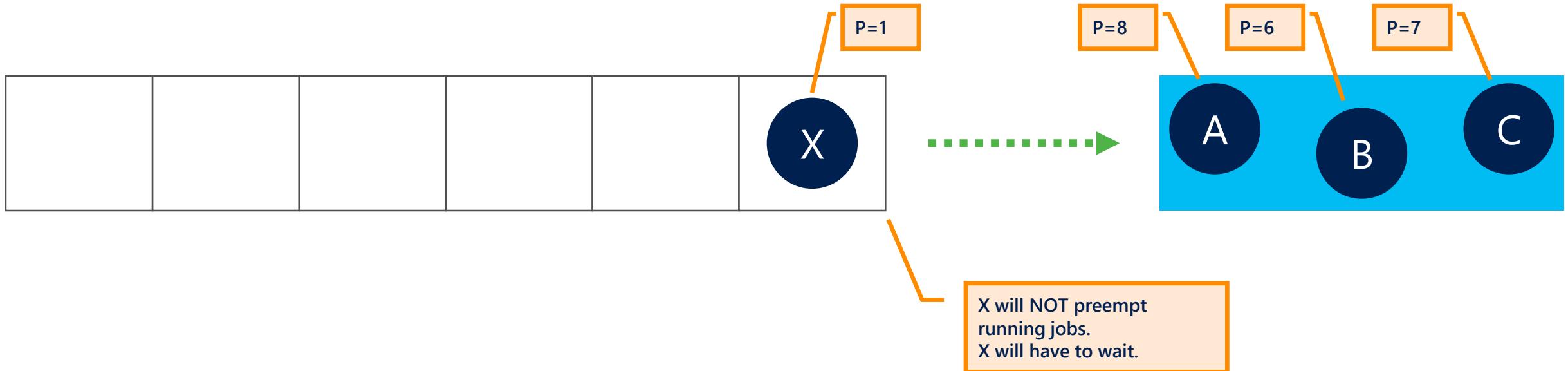
# Priority controls queue order



# Jobs only start if all requested containers can be reserved



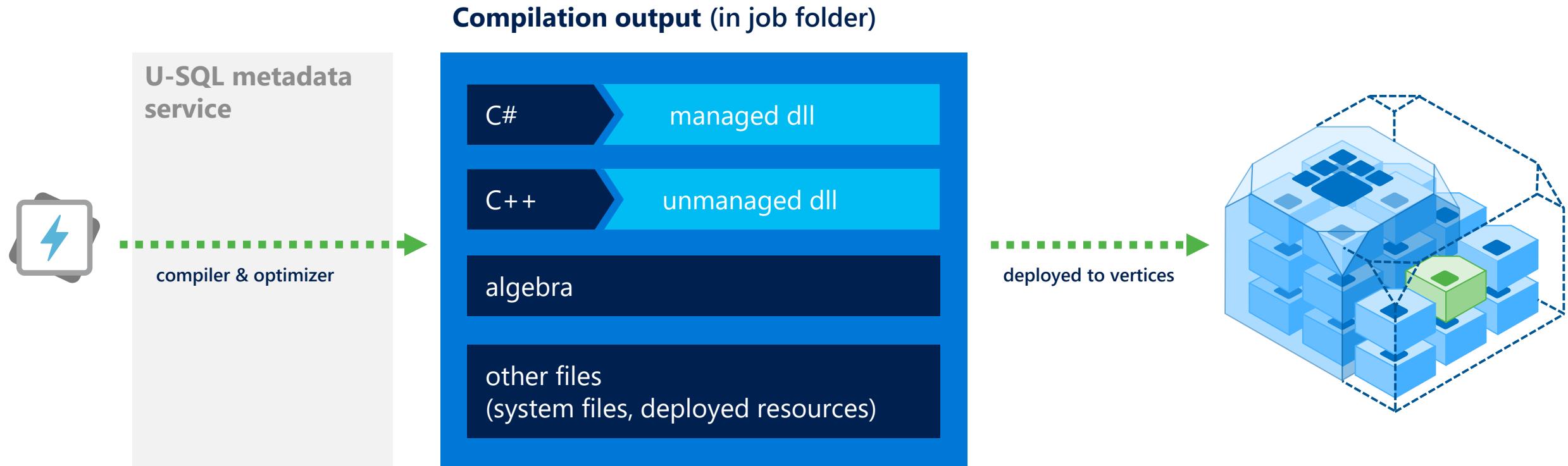
# Priority doesn't preempt running jobs



# U-SQL job compilation



# U-SQL compilation process



# The job folder

Inside the Default ADL Store:

## General Form

/system/jobservice/jobs/Usql/  
YYYY/MM/DD/hh/mm/  
JOBID

## An Example

/system/jobservice/jobs/Usql/  
2016/01/20/00/00/  
17972fc2-4737-48f7-81fb-49af9a784f64

NAME	SIZE
_Ast_.bin	176 KB
_scope_statlines_all.stp	
_ScopeCodeGen_.dll	
_ScopeCodeGen_.dll.cs	19.5 KB
_ScopeCodeGen_.pdb	24.1 KB
_ScopeCodeGenCompileOptions_.txt	6.75 KB
_ScopeCodeGenCompileOutput_.txt	4.26 KB
_ScopeCodeGenEngine_.cppresources	415 KB
_ScopeCodeGenEngine_.dll	
_ScopeCodeGenEngine_.dll.cpp	
_ScopeCodeGenEngine_.pdb	9.19 MB
_ScopeRuntimeStatistics_.xml	112 KB
_SStreamInfo_.xml	
algebra.xml	
commonrejoblibraries.dll	37.9 KB
newtonsoft.json.dll	5.11 KB
profile	
request.script	6.06 KB
ScopeVertexDef.xml	

C# code generated by the U-SQL Compiler

C++ code generated by the U-SQL Compiler

Cluster Plan a.k.a. "Job Graph" generated by U-SQL Compiler

User-provided .NET Assemblies

User-provided U-SQL script



## Resource

USQLSampleApplication1 - Microsoft Visual Studio

File Edit View Project Build Debug Team Data\_Lake Tools Test Analyze Window Help Analytics Demo AD

Quick Launch (Ctrl+Q)

LoadAndProcessData\_Resources Job Browser: ntfindemoadla0630 Ambulance-3-2-CreatedPartitionedTable.usql Compile View: C:\...\yDriversOnADL.usql

Job Resources

Name

Name	Download	File Size	Created
<a href="#">Algebra.xml</a>	<a href="#">Download all the resources</a>	5,306 bytes	N/A
<a href="#">DailyTradesAssembly.dll</a>	<a href="#">Download</a>	4,608 bytes	N/A
<a href="#">ScopeCodeGen .dll</a>	<a href="#">Download</a>	17,408 bytes	N/A
<a href="#">ScopeCodeGen .pdb</a>	<a href="#">Download</a>	60,928 bytes	N/A
<a href="#">ScopeCodeGenEngine .dll</a>	<a href="#">Download</a>	2,990 KB	N/A
<a href="#">ScopeCodeGenEngine .pdb</a>	<a href="#">Download</a>	44,644 KB	N/A
<a href="#">ScopeVertexDef.xml</a>	<a href="#">Download</a>	115,834 bytes	N/A
<a href="#">SStructInfo .xml</a>	<a href="#">Download</a>	847 bytes	N/A
<a href="#">ScopeCodeGen .dll.cs</a>	<a href="#">Download</a>	41,260 bytes	N/A
<a href="#">ScopeCodeGenEngine .dll.cpp</a>	<a href="#">Download</a>	1,753 KB	N/A
<a href="#">ScopeCodeGenCompileOutput .txt</a>	<a href="#">Download</a>	376,654 bytes	N/A
<a href="#">ScopeCodeGenCompileOptions .txt</a>	<a href="#">Download</a>	20,960 bytes	N/A
<a href="#">ScopeCodeGenEngine .cppresources</a>	<a href="#">Download</a>	483,347 bytes	N/A

Download All

Blue items: the output of the compiler

Grey items: U-SQL runtime bits

Download all the resources

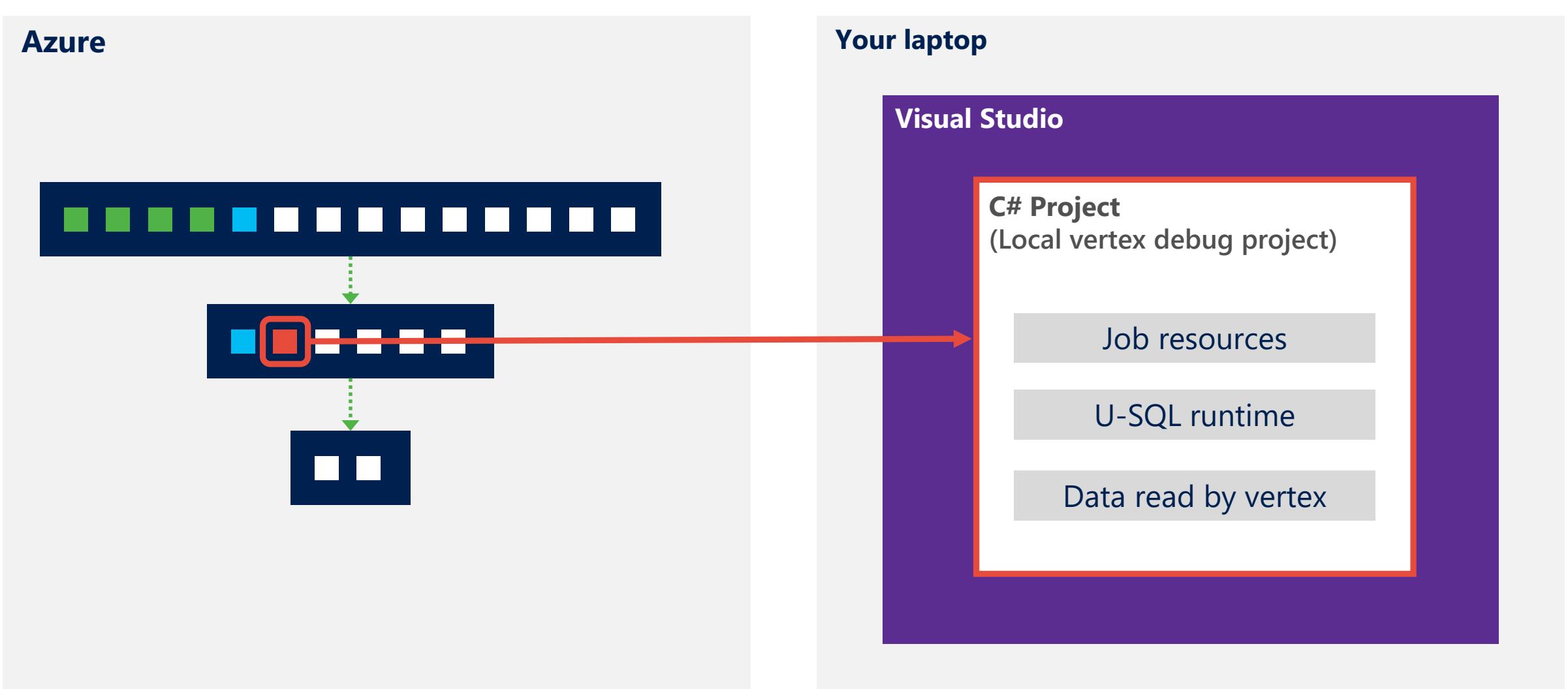
Download a specific resource

# Local vertex debug

(coming soon)

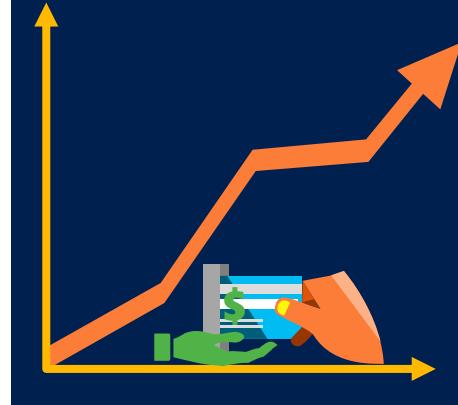


# Debug a vertex = run the vertex on your machine

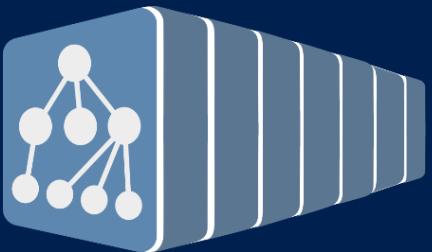


# Azure Data Lake Analytics

Start in seconds  
Scale instantly  
Pay per job



Develop massively parallel programs with simplicity



Debug and optimize your Big Data programs with ease



Virtualize your analytics



Enterprise-grade security, auditing and support

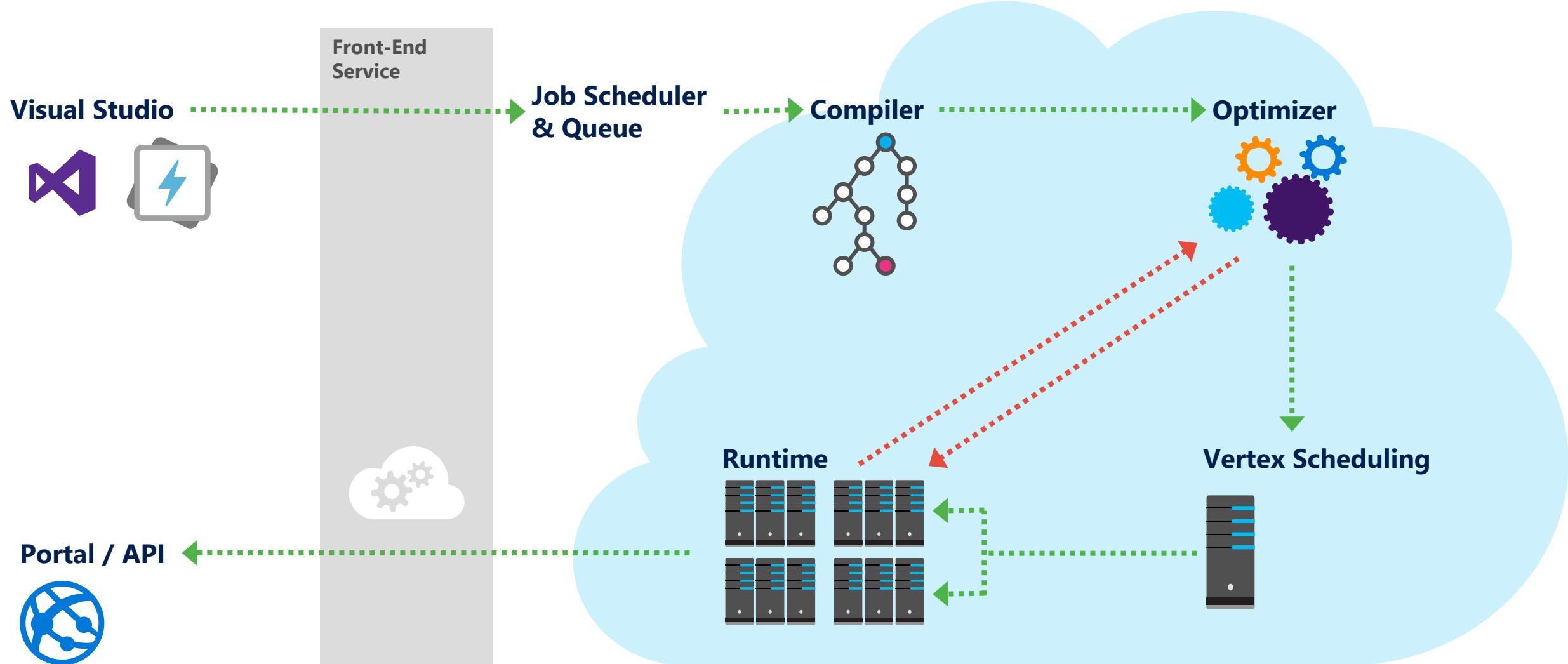


# Query execution

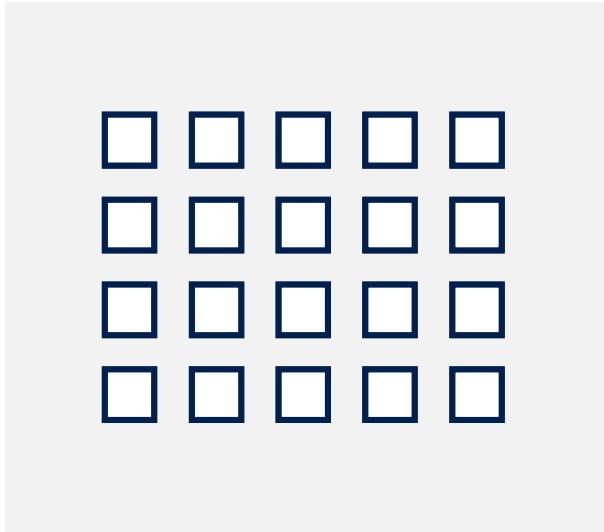
## Plans, vertices, stages, parallelism, ADLAUs



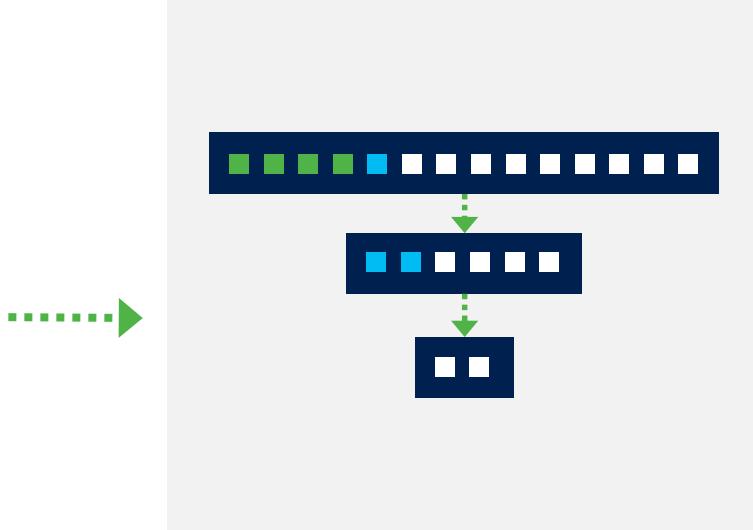
# Query Life



# Logical to physical plan

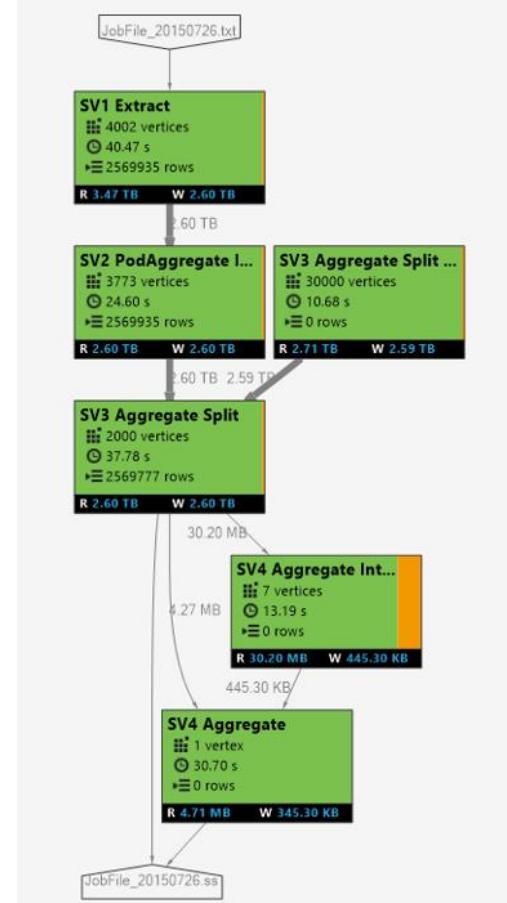


Each square = "a vertex"  
represents a fraction of  
the total



Vertices in each SuperVertex  
(aka "Stage") are doing the  
same operation on the same  
data.

Vertices in a later stages may  
depend on a vertex in an  
earlier stage

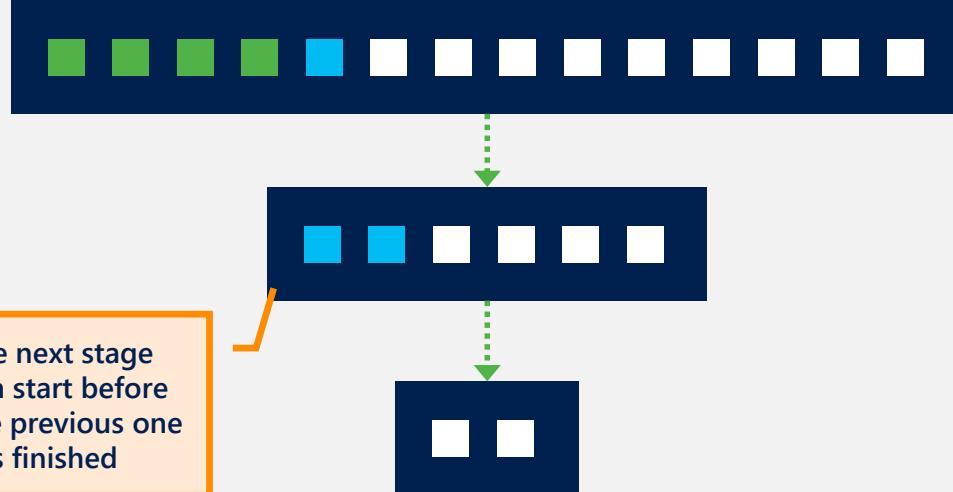


Visualized like this

# Execution with requested parallelism

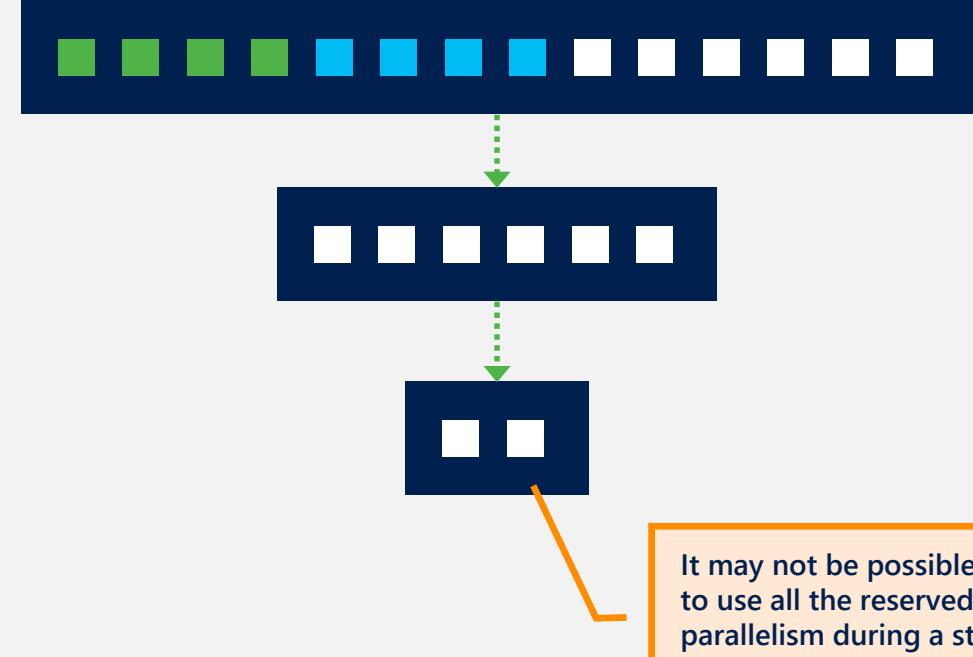
**Requested parallelism = 1**

(reserve enough to do 1 vertex at a time)



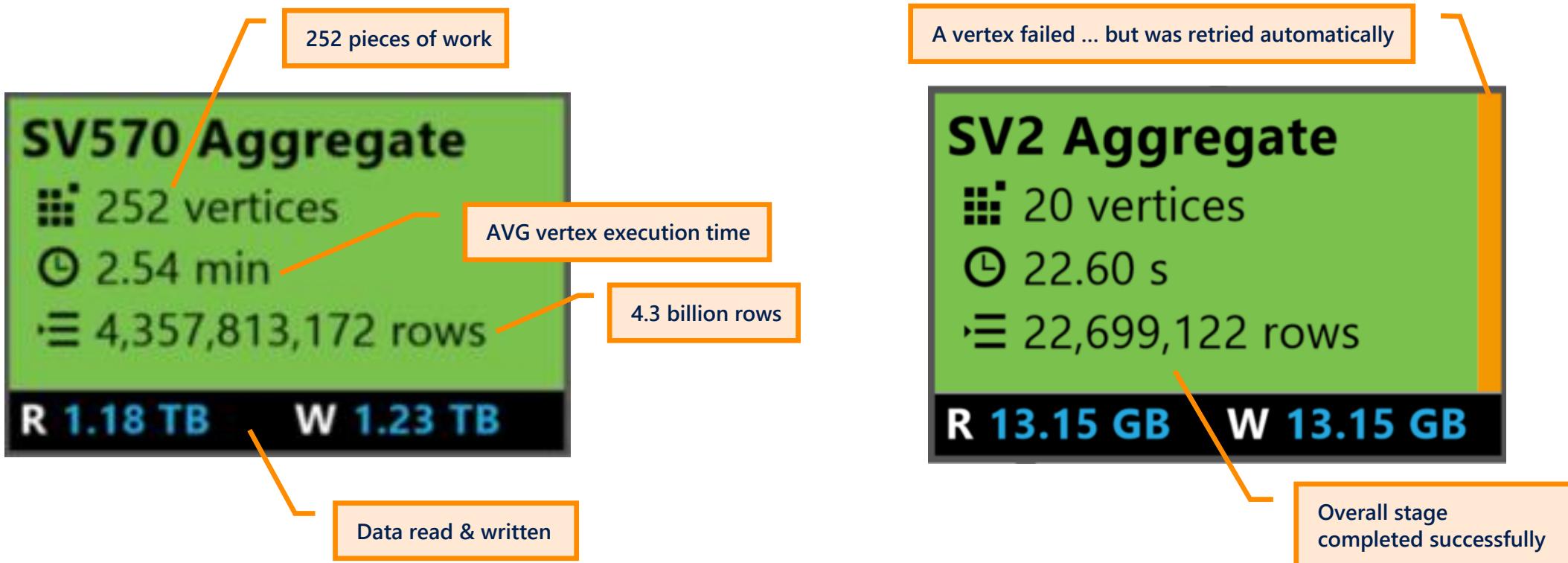
**Requested parallelism = 4**

(reserve enough to do 4 vertices at a time)



Job resources are copied to each vertex

# Stage details

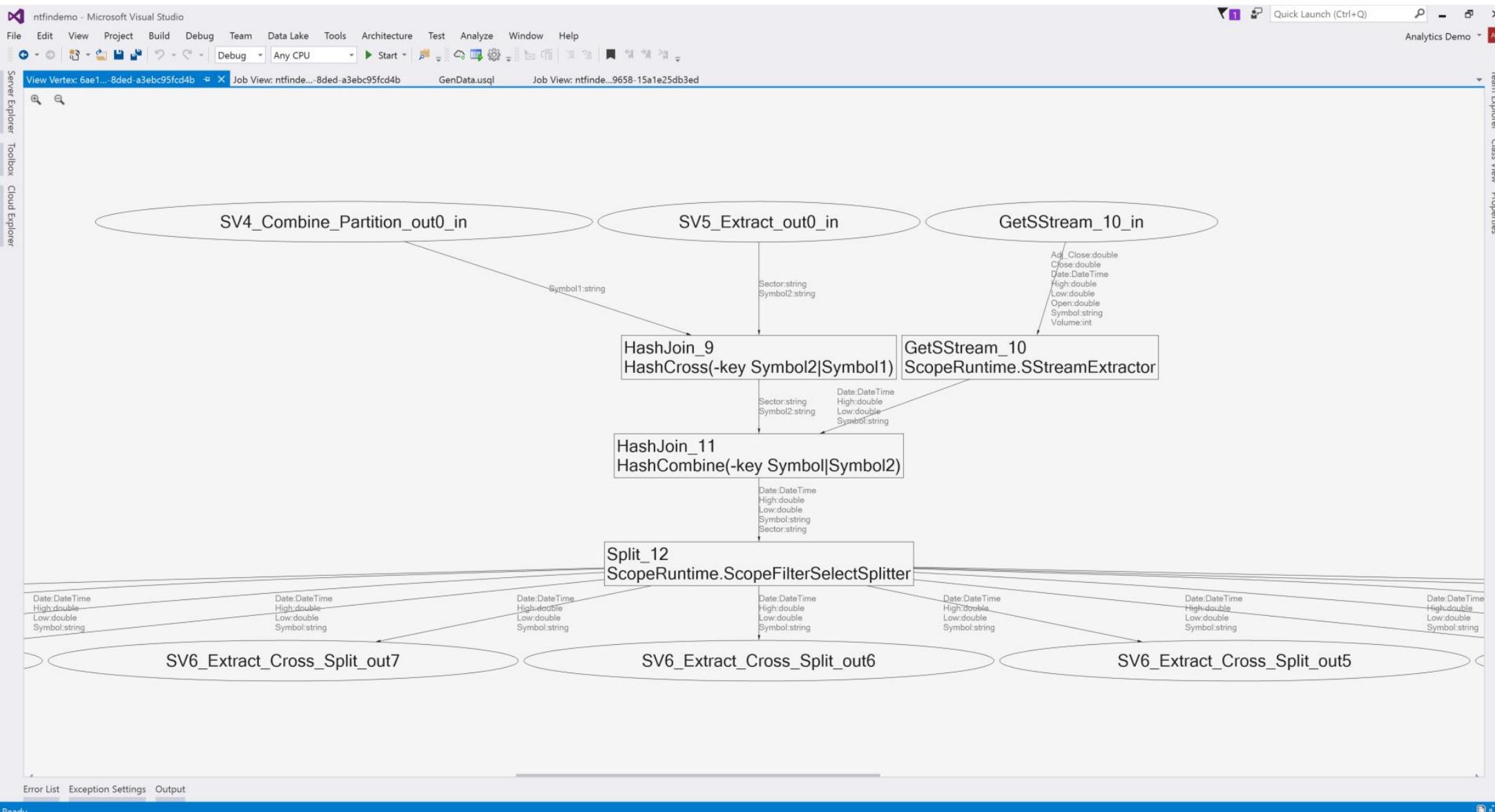


A vertex might fail because:

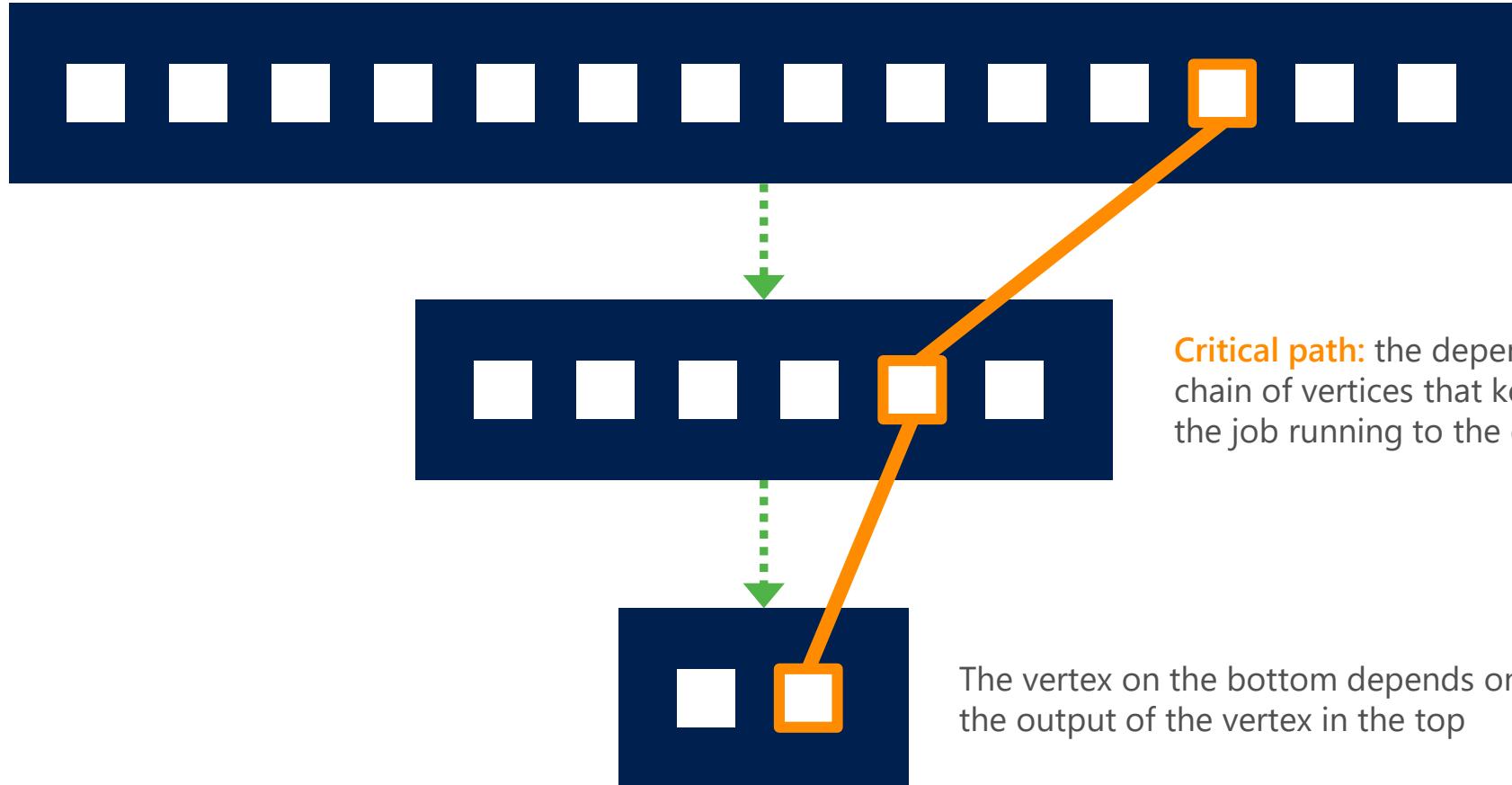
- Router congested
- Hardware failure (ex: hard drive failed)
- VM had to be rebooted

U-SQL job will automatically schedule a vertex on another VM.

# Inside a Stage: The Operator Graph



# Vertex relationships



ntfindemo - Microsoft Visual Studio

File Edit View Project Build Debug Team Data Lake Tools Architecture Test Analyze Window Help

Quick Launch (Ctrl+Q)

Analytics Demo AD

Server Explorer Toolbox Cloud Explorer

Job View: ntfinde... 8ded-a3ebc95fcdb Job View: ntfinde... 9658-15a1e25db3ed

Job Name: LoadAndProcessData | Load Profile

**Job Summary**

- Preparing (✓)
- Queued (✓)
- Running (✓)
- Finalizing (✓)

35 seconds 0 seconds 7.3 hours

**Job Completed Successfully**  
No errors to report.

**Job Result** Succeeded  
**Total Duration** 7.3 hours  
**Submit Time** 11/7/2016 9:45:27 AM  
**Start Time** 11/7/2016 9:46:20 AM  
**End Time** 11/7/2016 5:03:00 PM  
**Compilation** 35 seconds  
**Queued** 0 seconds  
**Running** 7.3 hours  
**Account** ntfindemoada0630  
**Author** analyticsdemo@outlook.com  
**Priority** 1000  
**Parallelism** 60  
**Bytes Left** 759,092,887,081  
**Bytes Read** 4,390,223,643,208  
**Bytes Written** 7,340,221,434,900  
**Vertices** 1,953

**Job Graph** **Metadata Operations** **State History** **Diagnostics**

Display: Progress Succeeded Failed Running Waiting

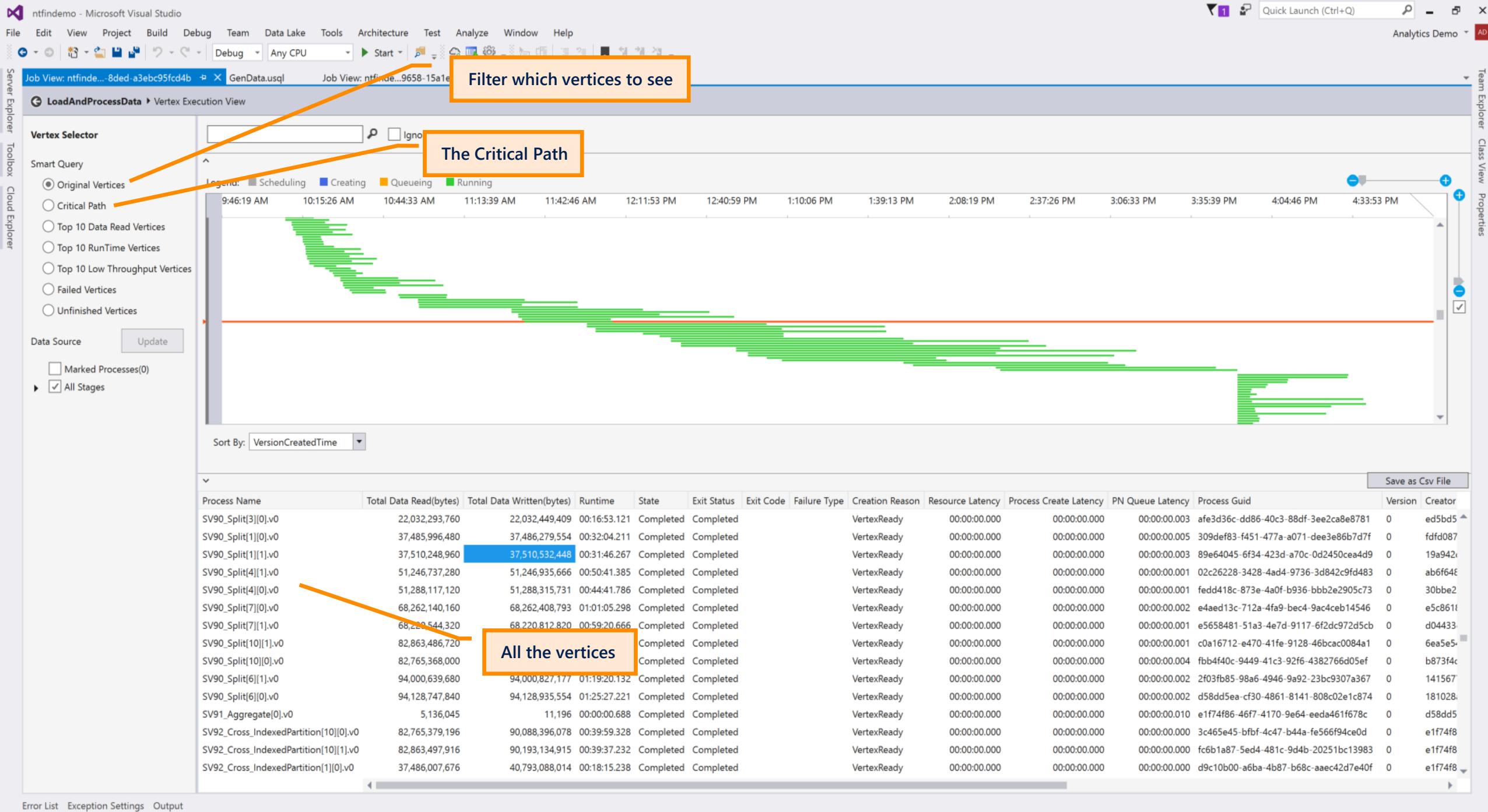
**Vertex Execution View**

Script Resources Vertex Execution View

Job Playback 00:00:00

Error List Exception Settings Output

Ready



# Efficiency

## Cost vs. latency



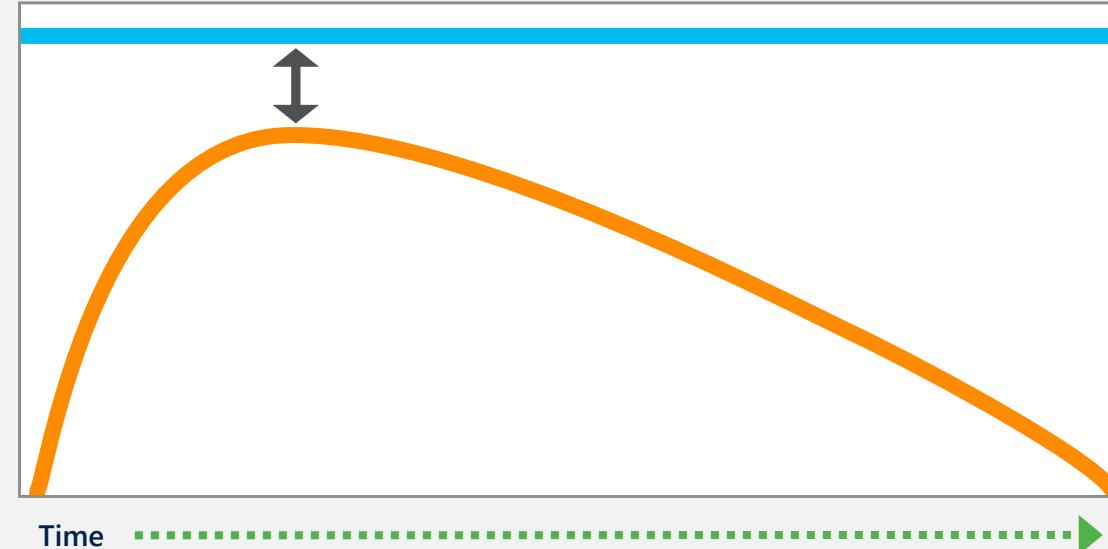
# ADLAU allocation

Example: allocating 10 ADLAUs for a 10 minute job

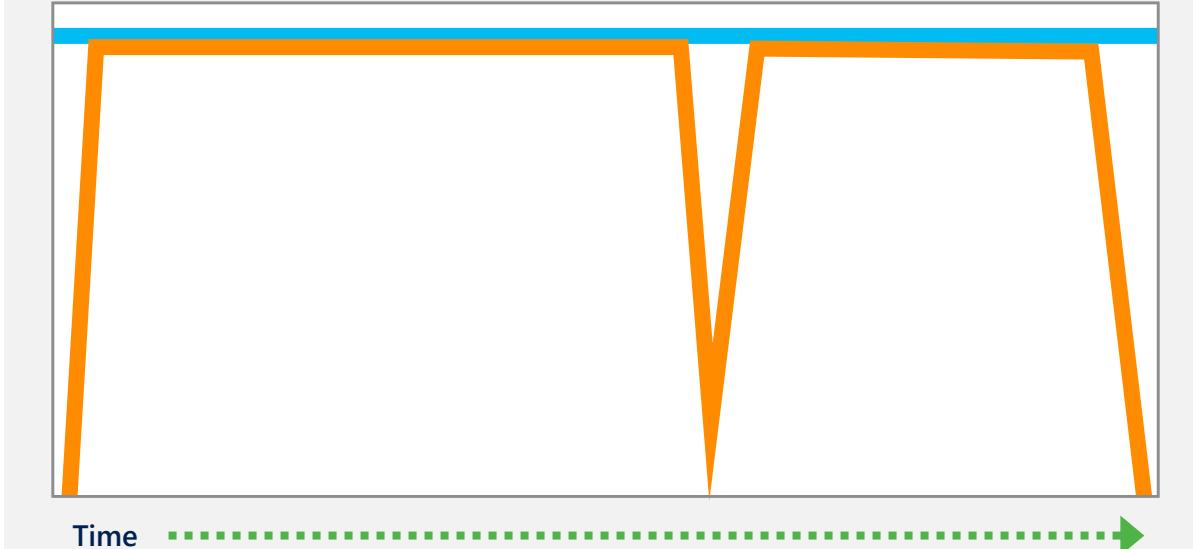
Cost:  $10 \text{ min} * 10 \text{ ADLAUs} = 100 \text{ ADLAU minutes}$

Blue line: allocated  
Red line: running

## Over-allocation



## Under-allocation



# Azure Data Lake Analytics

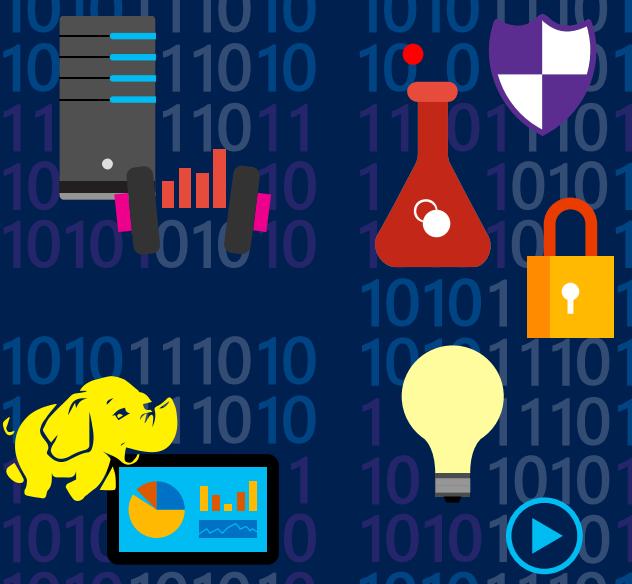
*Compare with HDInsight*



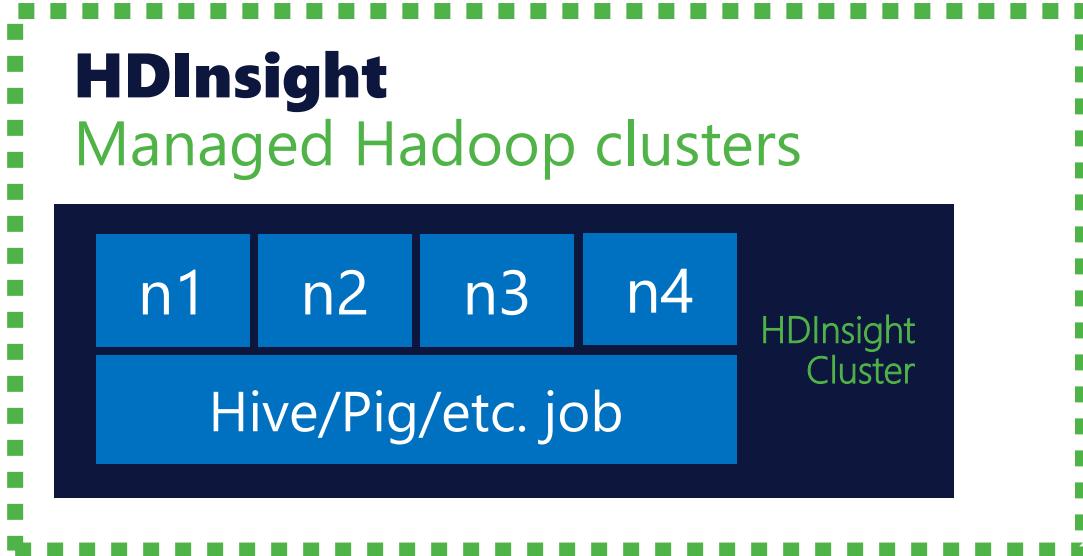
10101010101110111011  
1010101010101010111010  
10101010101110111011

1010111010 101011101  
10111010 101011101  
11110111 11101101  
10101010 10101101  
10101010 10101101

1010111010 101011101  
10111010 10101101  
11110111 11101101  
10101010 10101101  
10101010 10101101



# Analytics: Two form factors



# ADLA complements HDInsight

## Target the same scenarios, tools, and customers

### **HDInsight**

- ⚡ For developers familiar with the Open Source: Java, Eclipse, Hive, etc.
- ⚡ Clusters offer customization, control, and flexibility in a managed Hadoop cluster

### **ADLA**

- ⚡ Enables customers to leverage existing experience with C#, SQL & PowerShell
- ⚡ Offers convenience, efficiency, automatic scale, and management in a “job service” form factor

# Choose the right service

## Decision factor

## Example

Capability

Hadoop, Spark vs U-SQL

Control

G-Series VMs  
Customization scripts

Ecosystem

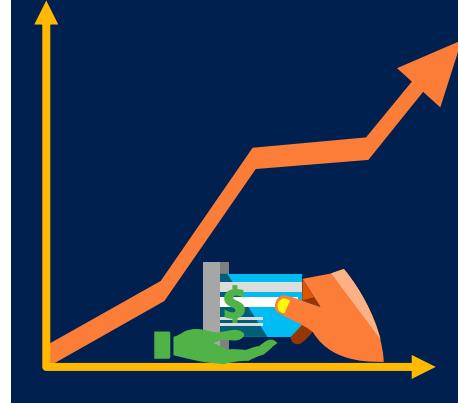
DataMeer, AtScale, Cask etc.

Form Factor

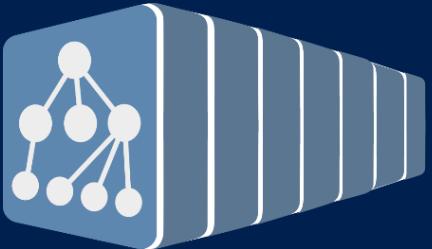
Job Service Vs Clusters

# Azure Data Lake Analytics

Start in seconds  
Scale instantly  
Pay per job



Develop massively parallel programs with simplicity



Debug and optimize your Big Data programs with ease



Virtualize your analytics

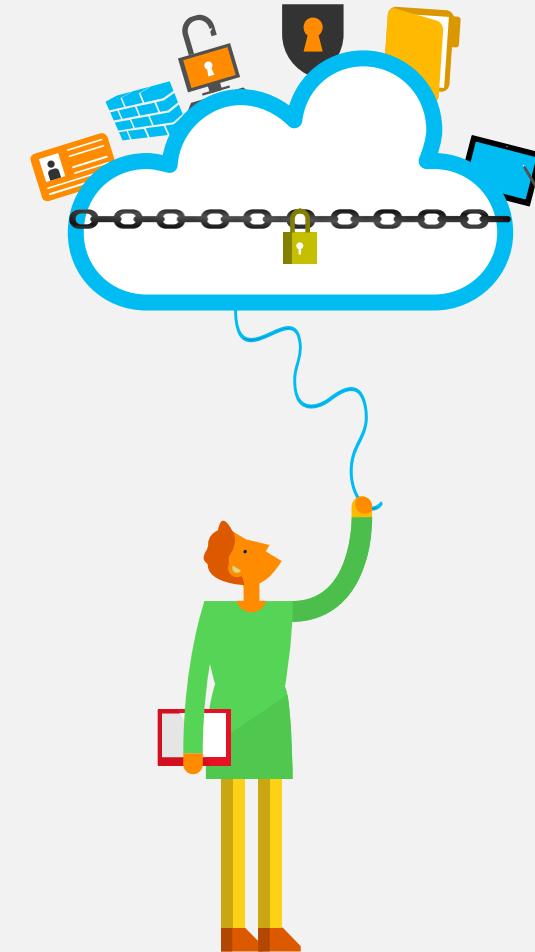


Enterprise-grade security, auditing and support



# ADL Store: enterprise-grade security

- ⚡ Enterprise-grade security permits even sensitive data to be stored securely
- ⚡ Regulatory compliance can be enforced
- ⚡ Integrates with Azure Active Directory for authentication
- ⚡ Data is encrypted at rest
- ⚡ POSIX-style permissions on files and folders
- ⚡ Audit logs for all operations



# ADL Store Security: AAD integration

- ⚡ Multi-factor authentication based on OAuth2.0
- ⚡ Integration with on-premises AD for federated authentication
- ⚡ Self-service password management
- ⚡ Role-based access control
- ⚡ Privileged account management
- ⚡ Device registration
- ⚡ Application usage monitoring and rich auditing
- ⚡ Security monitoring and alerting
- ⚡ Fine-grained ACLs for AD identities



# ADL Store security: Role-based access

- ⚡ Each file and directory is associated with an owner and a group
- ⚡ Files or directories have separate permissions (read(r), write(w), execute(x)) for owners, members of the group, and for all other users
- ⚡ Fine-grained access control lists (ACLs) rules can be specified for specific named users or named groups

The screenshot shows the Microsoft Azure portal interface for managing access to a resource. The top navigation bar includes a search bar, notification bell, edit, gear, help, and Microsoft logo. Below the header, the page title is "Access /". There are "Add", "Save", and "Discard" buttons. The main content area is titled "Standard Access" and displays a table of permissions:

	NAME	READ	WRITE	EXECUTE
Owner	@outlook.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Group	@outlook.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Other	All Users and Groups	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

# ADL Store Security: encryption-at-rest

- ⚡ Transparently encrypts data flowing to and from public networks as well as at rest
- ⚡ Transparent server-side encryption
- ⚡ User can manage their own encryption keys or let Azure Data Lake Store manage the key using Azure Key Vault





# Get started today!



For more information visit:  
<http://azure.com/datalake>



