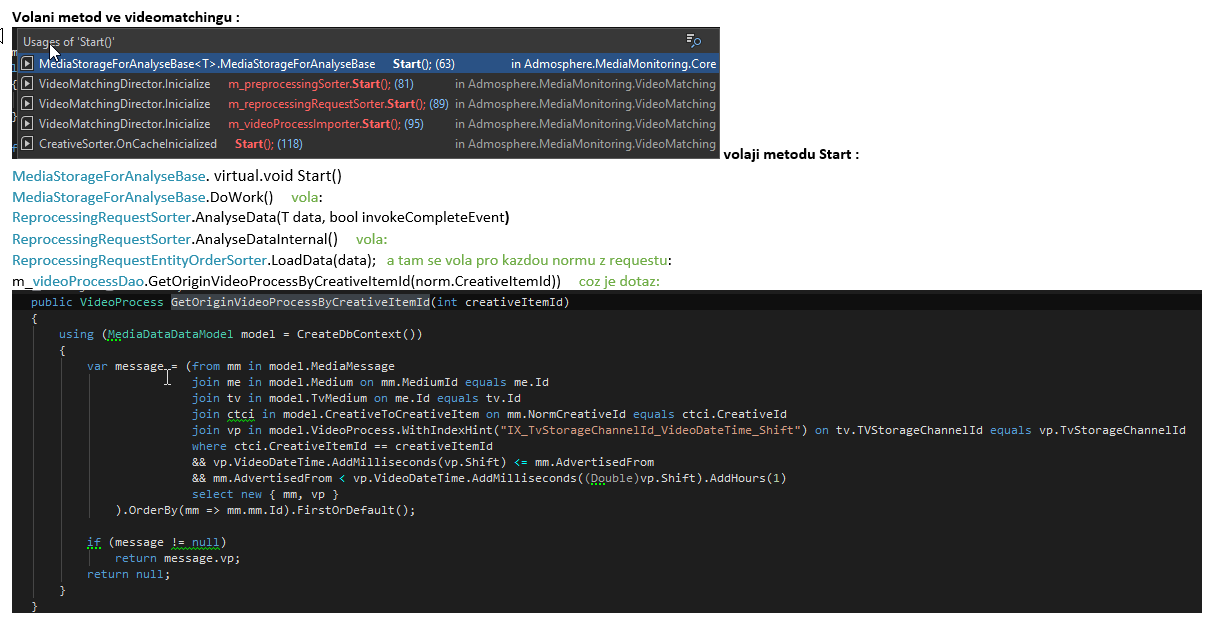
[Bug 51346](https://tfs.mediaresearch.cz/AdIntel/AdIntel/_workitems/edit/51346): VM - timeoutuje zpracovávání některých reprocessing requestů





# Test bez otoceni indexu:

**MediaData3Auto (se spatne otocenym indexem)**:

**Puvodni query:**

Prvni spuseni: 38 sekund

Druhe spusteni: 6 sekund

**Nova query:**

Prvni spuseni: 2 minuty 40 sekund

Druhe spusteni: 2 minuty 38 sekund

**Alfred (se spatne otocenym indexem):**

**Puvodni query:**

Prvni spuseni: 4 sekundy

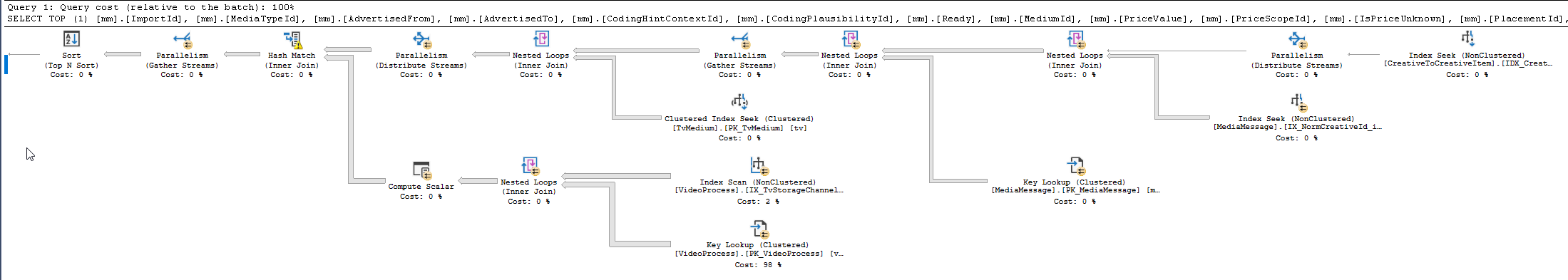
Druhe spusteni: 3 sekundy

Nova query:

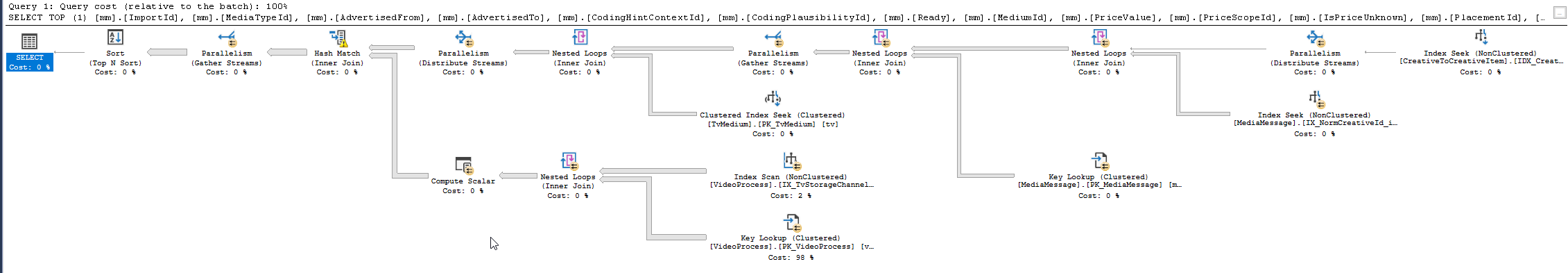
Prvni spuseni: 2 minuty 53 sekund

Druhe spusteni: 2 minuty 59 sekund (s exekucnim planem)

**Exekucni plan puvodni query:**

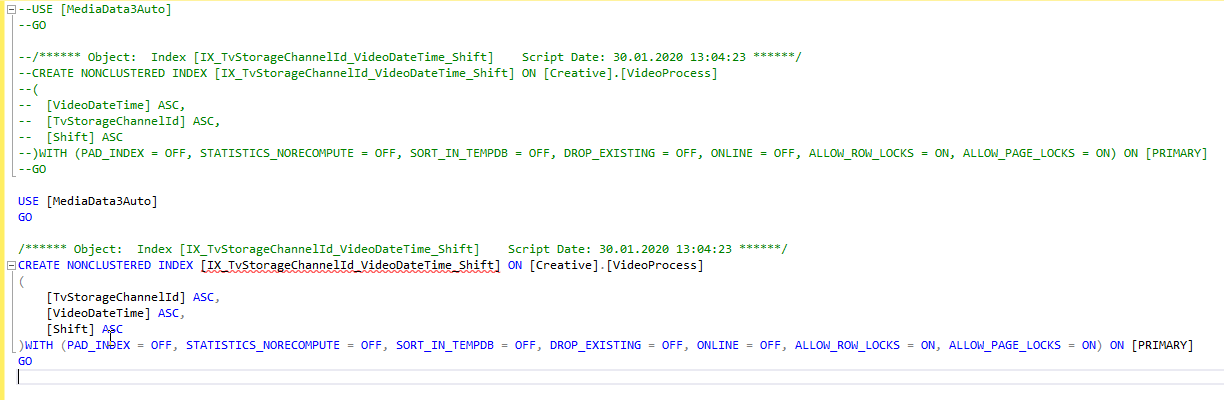


**Exekucni plan nova query:**

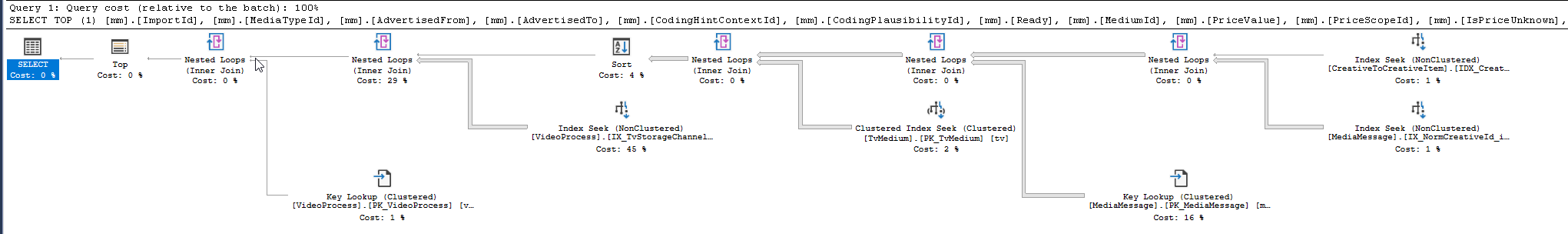


# Test po otoceni indexu:

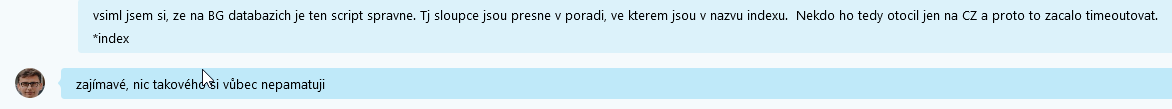
**Kdyz zmenim index prohozenim channel a videoDateTime (Puvodni index je zakomentovany), dostanu se pod jednu sekundu pro obe query (jiz nacacheovana data) i na testovaci databazi :**



**Exekucni plan po zmene indexu:**



# Reseni:



1. **Otocil jsem index (na CZ byl obracene, spravne poradi sloupcu je tak, jak je v nazvu indexu):**

USE [MediaData3BG]

GO

/\*\*\*\*\*\* Object: Index [IX\_TvStorageChannelId\_VideoDateTime\_Shift] Script Date: 30.01.2020 15:30:04 \*\*\*\*\*\*/

DROP INDEX [IX\_TvStorageChannelId\_VideoDateTime\_Shift] ON [Creative].[VideoProcess]

GO

/\*\*\*\*\*\* Object: Index [IX\_TvStorageChannelId\_VideoDateTime\_Shift] Script Date: 30.01.2020 15:30:04 \*\*\*\*\*\*/

CREATE NONCLUSTERED INDEX [IX\_TvStorageChannelId\_VideoDateTime\_Shift] ON [Creative].[VideoProcess]

(

[TvStorageChannelId] ASC,

[VideoDateTime] ASC,

[Shift] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

GO

1. **Rozdelil jsem query:**

public VideoProcess GetOriginVideoProcessByCreativeItemId(int creativeItemId)  
{  
 using (MediaDataDataModel model = CreateDbContext())  
 {  
 var message = (from mm in model.MediaMessage  
 join ctci in model.CreativeToCreativeItem on mm.NormCreativeId equals ctci.CreativeId  
 where ctci.CreativeItemId == creativeItemId  
 select mm).OrderBy(mm => mm.Id).FirstOrDefault();  
   
 if (message == null)  
 return null;  
  
 var videoProcess = (from vp in model.VideoProcess  
 join tvMe in model.TvMedium on vp.TvStorageChannelId equals tvMe.TVStorageChannelId  
 where tvMe.Id == message.MediumId &&  
 vp.VideoDateTime.AddMilliseconds(vp.Shift) <= message.AdvertisedFrom &&  
 message.AdvertisedFrom < vp.VideoDateTime.AddMilliseconds(vp.Shift).AddHours(1)  
 select vp).FirstOrDefault();  
  
 return videoProcess;  
 }  
}

1. **Odchycene dotazy po rozdeleni (rychlost pod 1 s):**

exec sp\_executesql N'SELECT TOP (1)

[mm].[Id],

[mm].[AdvertisedFrom],

[mm].[MediumId]

FROM

[Media].[MediaMessage] [mm]

INNER JOIN [Creative].[CreativeToCreativeItem] [ctci] ON [mm].[NormCreativeId] = [ctci].[CreativeId]

WHERE

[ctci].[CreativeItemId] = @creativeItemId

ORDER BY

[mm].[Id]

',N'@creativeItemId int',@creativeItemId=10572388

--

exec sp\_executesql N'SELECT TOP (1)

[vp].[TvStorageOutputRequestId],

[vp].[VideoProcessStatusId],

[vp].[Modified],

[vp].[ModifiedBy],

[vp].[Created],

[vp].[CreatedBy],

[vp].[VdtDataIsImported],

[vp].[VdtDataExist],

[vp].[PeaksDataExist],

[vp].[MatchingTypeReady],

[vp].[VideoDateTime],

[vp].[Shift],

[vp].[VideoLength],

[vp].[TvStorageChannelId],

[vp].[TvStorageLocationId],

[vp].[Id]

FROM

[Creative].[VideoProcess] [vp]

INNER JOIN [Media].[TvMedium] [tvMe] ON [vp].[TvStorageChannelId] = [tvMe].[TVStorageChannelId]

WHERE

[tvMe].[Id] = @p1 AND DateAdd(Millisecond, Convert(Float, [vp].[Shift]), [vp].[VideoDateTime]) <= @AdvertisedFrom AND

@AdvertisedFrom1 < DateAdd(Hour, 1, DateAdd(Millisecond, Convert(Float, [vp].[Shift]), [vp].[VideoDateTime]))

',N'@p1 int,@AdvertisedFrom datetime,@AdvertisedFrom1 datetime',@p1=494,@AdvertisedFrom='2018-06-04 10:10:25.520',@AdvertisedFrom1='2018-06-04 10:10:25.520'

**#51346 Videomatching - rozdelil jsem query na dva dotazy kvuli timeoutu. Po rozdeleni oba dotazy dobehnou v radu milisekund.**

**Chapu to takto:**

**Holka kdyz strihne reklamu, inprocessing ji projede ten stream ve kterem to strihla az do konce. Reprocessing ale projede vsechny hodiny (i ty, ktere nejdou do Zachytavadla) proti teto nove norme.**

**Tento dotaz vrati videoProces ve kterem je urcity CreativeItem na zaklade CreativeItemId**

**~~Query bylo potreba zoptimalizovat tak jako tak..~~**

**Proto si myslim, ze ten dotaz chceme pokladat.**

Mam nejaky ReprocessingRequest . Ten ma na sobe normy. Kazda norma, ma jedno creativeItemId.

Kdyz holka otevre hodinu, chce aby se ji videomatching podival, jestli na te hodine neprobiha nejaky reprocessing.

* Co jsem vykoukal v Z. – pozadavek na reprocessing vznikne pri vystrizeni zpravy. Videomatchingu se predava MM.Id. VM si musi nejak najit normu a z ni creativeItemId.
* Pri otevreni streamu by se mel volat preprocessing, to jsem nenasel, ale do nasi dao metody preprocessing requesty taky dorazi – viz volani metod.

