|  |
| --- |
| Erik Darling says: [Anatomy Of An Adaptive Join](http://brentozar.us2.list-manage.com/track/click?u=9082566fb63d87be35c0662bc&id=fdb6d43c3e&e=b137e91a2b)I don’t like it unless it’s brand new When new features drop, not everyone has time to jump on top of them and start looking at stuff. That’s what consultants with nothing better to do are for.  I’ve been excited about this feature since talking to The Honorable Joseph Q. Sack, Esq. about it at PASS last October. My pupils dilated like I just found the bottom of a bottle of Laphroaig 18.  Let’s look at the Adaptive Join process in a way that most of you will eventually see it: in a query plan. Query Plan. Should I capitalize that? The jury is out. The Operator This is what the operator itself looks like:  [https://www.brentozar.com/wp-content/uploads/2017/12/16-2017-04-27_17-27-26.jpg](http://brentozar.us2.list-manage.com/track/click?u=9082566fb63d87be35c0662bc&id=5def87362c&e=b137e91a2b)  Greased Lightning    And as of CTP2 of 2017, this is the information available in the Adaptive Join operator itself.  [https://www.brentozar.com/wp-content/uploads/2017/04/14-2017-04-27_16-55-59.jpg](http://brentozar.us2.list-manage1.com/track/click?u=9082566fb63d87be35c0662bc&id=d8d96b7dca&e=b137e91a2b)  Evolution  Some points of interest:   * Actual Join Type: doesn’t tell you whether it chose Hash or Nested Loops * Estimated Join Type: Probably does * Adaptive Threshold Rows: If the number of rows crosses this boundary, Join choice will change. Over will be Hash, under will be Nested Loops.   The rest is fairly self-explanatory and is the usual stuff in query plans. The Plan Each Adaptive Join plan will have a branch for each path the optimizer can choose.  In our case, the Index Scan/Hash Join plan is the first path, and the Index Seek/Nested Loops Join is the second path.  [https://www.brentozar.com/wp-content/uploads/2017/04/15-2017-04-27_17-02-49.jpg](http://brentozar.us2.list-manage2.com/track/click?u=9082566fb63d87be35c0662bc&id=75fd08a928&e=b137e91a2b)  Hekaton doesn’t even have query plans  Inside the plan, there are a couple visual cues that let you know which path the query took.   1. The width of the lines: the wider line likely had data flow through it 2. The number of executions line in the tool tip for the index access: So far as I’ve seen, the path the query chooses will have > 0, and the path the query didn’t choose will have 0 here.   [https://www.brentozar.com/wp-content/uploads/2017/12/06-2017-04-27_17-23-46.jpg](http://brentozar.us2.list-manage.com/track/click?u=9082566fb63d87be35c0662bc&id=c233091309&e=b137e91a2b)  Best Execution Ever. Misc In other posts I’ve mentioned stuff! And things. I’m going to bring it all together here.   1. You need CTP2 of SQL Server 2017 to use Adaptive Joins (for now) 2. At least one table has to use a ColumnStore index (for now) 3. Database compatibility has to be 140 (for now) 4. You need to be using SSMS 2017 to see the Adaptive Join operator in graphical query plans (you can see it in SET STATISTICS XML ON otherwise, it’s text only) 5. Right now, only Nested Loops and Hash Join are supported (my best guess is that Merge Join was left out because of the potential cost of having to inject a Sort operator into the plan)   Whew. Hey. That’s it. Go home.  Thanks for reading!  **Brent says:** this has interesting implications for index-tracking DMVs like [sys.dm\_db\_index\_usage\_stats](http://brentozar.us2.list-manage1.com/track/click?u=9082566fb63d87be35c0662bc&id=89d1949420&e=b137e91a2b). In the past, I’ve explained that DMV as “counting the number of times a plan with that index in it has executed,” but I’m going to have to dig deeper now that it shows the same index twice here with two different operations (a seek and a scan.) [Read the comments.](http://brentozar.us2.list-manage.com/track/click?u=9082566fb63d87be35c0662bc&id=348834f6ce&e=b137e91a2b) |