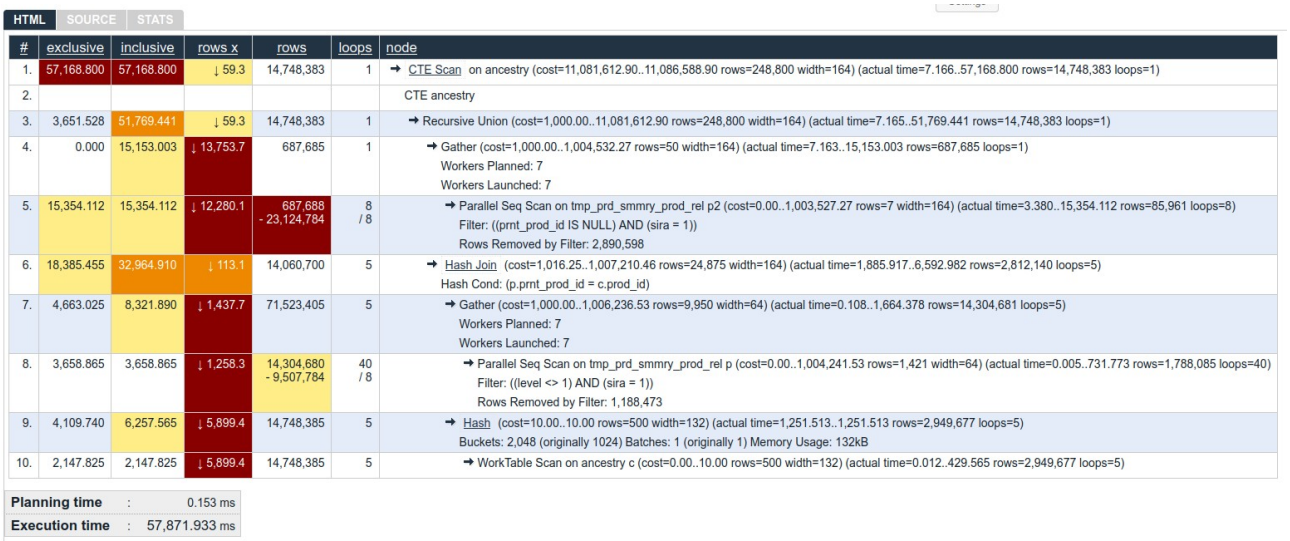
Ref: https://explain.depesz.com/help



HTML | SOURCE | STATS

#	exclusive	inclusive	rows x	rows	loops	node
1.	57,168.800	57,168.800	↓ 59.3	14,748,383	1	→ CTE Scan on ancestry (cost=11,081,612.90..11,086,588.90 rows=248,800 width=164) (actual time=7.166..57,168.800 rows=14,748,383 loops=1)
2.						CTE ancestry
3.	3,651.528	51,769.441	↓ 59.3	14,748,383	1	→ Recursive Union (cost=1,000.00..11,081,612.90 rows=248,800 width=164) (actual time=7.165..51,769.441 rows=14,748,383 loops=1)
4.	0.000	15,153.003	↓ 13,753.7	687,685	1	→ Gather (cost=1,000.00..1,004,532.27 rows=50 width=164) (actual time=7.163..15,153.003 rows=687,685 loops=1) Workers Planned: 7 Workers Launched: 7
5.	15,354.112	15,354.112	↓ 12,280.1	687,688 - 23,124,784	8 / 8	→ Parallel Seq Scan on tmp_prd_smmry_prod_rel p2 (cost=0.00..1,003,527.27 rows=7 width=164) (actual time=3.380..15,354.112 rows=85,961 loops=8) Filter: ((prnt_prod_id IS NULL) AND (sira = 1)) Rows Removed by Filter: 2,890,598
6.	18,385.455	32,964.910	↓ 113.1	14,060,700	5	→ Hash Join (cost=1,016.25..1,007,210.46 rows=24,875 width=164) (actual time=1,885.917..6,592.982 rows=2,812,140 loops=5) Hash Cond: (p.prnt_prod_id = c.prod_id)
7.	4,663.025	8,321.890	↓ 1,437.7	71,523,405	5	→ Gather (cost=1,000.00..1,006,236.53 rows=9,950 width=64) (actual time=0.108..1,664.378 rows=14,304,681 loops=5) Workers Planned: 7 Workers Launched: 7
8.	3,658.865	3,658.865	↓ 1,258.3	14,304,680 - 9,507,784	40 / 8	→ Parallel Seq Scan on tmp_prd_smmry_prod_rel p (cost=0.00..1,004,241.53 rows=1,421 width=64) (actual time=0.005..731.773 rows=1,788,085 loops=40) Filter: ((level <> 1) AND (sira = 1)) Rows Removed by Filter: 1,188,473
9.	4,109.740	6,257.565	↓ 5,899.4	14,748,385	5	→ Hash (cost=10.00..10.00 rows=500 width=132) (actual time=1,251.513..1,251.513 rows=2,949,677 loops=5) Buckets: 2,048 (originally 1024) Batches: 1 (originally 1) Memory Usage: 132kB
10.	2,147.825	2,147.825	↓ 5,899.4	14,748,385	5	→ WorkTable Scan on ancestry c (cost=0.00..10.00 rows=500 width=132) (actual time=0.012..429.565 rows=2,949,677 loops=5)

Planning time	:	0.153 ms
Execution time	:	57,871.933 ms

Bazı önemli temel noktalar:

- **her düğüm ==> 1 operasyon**

- **Operasyonların işlenme sırası top-down; fakat kayıtlar üzerinde operasyonların işlenme sırası bottom-up !**

- **Planning time: Plannner plan üretmek icin harcadığı zaman.**
  Execution Time: Uretilen planın gerçekten çalışma zamanı.
  - Planning time ile Execution time arasında bir bağıntı YOK.

- **op_name (cost estimates) (actual values)**
  - tahmin kısmındaki değerler maliyet kestirimi. Birimi artificial. Genel olarak seq scan'da bir sayfa erişimi için maliyet 1.
  - gerçek değerlerdeki zaman birimi msec.
  - Her iki kısımdaki değerler sadece bir loop için ortalama değer. Eğer düğüm çok kez çalışırsa (loop=… ile ifade ediliyor), o düğüm için toplam değeri loop ile çarpmak gerek. Fakat bu durum paralel çalışmada biraz daha karışabiliyor. Aşağıda örnek var.

- **Actual time(start-up time, total exec time for this node (inclusive))**
  - start-up time ilk kayıt üretmek icin geçen zaman, yani hazırlanma zamanı. Sort, Hash gibi bazı operasyonlar ilk aşamaları partition iterasyonları içeriyor, ki bunlar zaman alıyor. Seq Scan gibi basit operasyonlar çok düşük bir gecikmeyle ilk kayda erişebiliyor.
  - Üst düğüm start-up time her zaman içerdiği alt-düğümlerin start-up time'larından daha yüksek olmalı. Bu Total-exec time için de böyle.
    - Sadece mevcut düğümün toplam gecikmesi için (yani exclusive), total-exec time'dan alt-düğümlerin total-exec timelerını çıkarmamız gerek.

- **rows :** düğümün ürettiği kayıt miktarı. (tahmin) (gerçek) Her ikisinde de var. Soldaki rows kolonunda bazen "-sayı" var. bu değer filtre içeren mevcut düğümün toplam remove ettiği kayıt sayısı.
- **rows x** : Düğümün ürettiği kayıt sayısı tahminindeki hata miktarı kaç kat (x)
- **En çok dikkat çeken noktalar kırmızı hücreler, yani dar boğazın en çok olduğu yerler.** Mesela,

- yüksek “rows x”. Mesela yukarıdaki örnekte 4,5,7,8,9,10
- “rows” kolonundaki “-sayı” değerinin çıkan kakıtlara göre çok yüksek olması. Mesela yukarıdaki örnekte 5. düğüm.
- Düğümün exclusive total-exec time, total-exec-time a oranının yüksek olması. Mesela, örnekte 1. düğüm.

Örnekler:

9, 10) 6257.565 ms = 1251.513 * 5 loops => 2048 sepetlik “hash table” kullanımı
WorkTable Scan on ancestry takes 2147.825 ms = 429.565 * 5
(WorkTable Scan genelde recursive CTE’de karşılaşılan bir şey!)
Sadece (exclusive) bu HASH işlemi 4109.740 ms alıyor.
Yani bir HASH yaklaşık 800 ms. (1251... – 429...)

rows x = 2949677 / 500 = 5899,4 kat aşağıda bir değer tahmin etmiş..büyük hata.
Rows = her bir loop’da ortalama kaç kayıt çıkadık.
9. düğüm her loop’da 2949677 kayıt çıkarıyor. Oyleyse toplam çıkan kayıt:
2949677 * 5 = 14748385

8) **B**urada tmp_.. tablosunda 7+1= 8 tane process paralel olarak seq scan yapıyor. Yani her biri dosyanın 1/8 inde çalışıyor. Her bir process’de 5 loop var. Toplam 40 loop oluyor. Her loop sonunda bir Gather islemi oluyor.
1 loop (paralel 8 process) = 731.773 ms
731.773 * 5 = 3658,865 (exlc. ve incl.) toplam sure.

Bir worker processin ortaya çıkardığı toplam kayıt sayısı : 1788085
8 tane worker = 8 * 1788085 = 14304680 tane kayıt toplam ortaya çıkan
Her worker process ortalama 1188473 remove ediyor. Toplam -9507784 remove ediliyor.

7) 1664,378×5 = 8321,89 ms

6) 8321,89+6257,565+18385,455 = 32964,91 ms

5) 15354,112 *1 → 8 process paralel filter
Her bir process 28905598 tane filtreliyor. Toplam 28905598 * 8 = 23124784

4) 15153, 003 ms da gather sonlanıyor.

3) 32 + 15 +3 = 51,...