	Cost	Output	Cost	Operator	Notes	IM		
				No network, AsterixDB Architecture			Assumption 3S	No network, AsterixDB Architecture
0A	E#$_f$F	#$_f$(|F|/V(c,F))	|E|(|F|/V(c,F))	Index Scan		F	w : e.c = f.c	V(C,F) >= V(C,E)
0C	sort(0A)	#$_f$V(g,F)/V(c,F)	0	in-memory sort		T	1/V(c,F) < Mf$_s$/|F|	E very small
0E	sort(0C)	min(#$_f$(V(g,F)),l)	0	in-memory sort	output will be l in our case	T		F very large
				No network, AsterixDB Architecture			Assumption 3M	No network, AsterixDB Architecture
0A	E#$_f$F	#$_f$(|F|/V(c,F))	E(F/V(c,F))	Index Scan		F	w : e.c = f.c	V(C,F) >= V(C,E)
0C	sort(0A)	#$_f$V(g,F)/V(c,F)	2#f(F/V(c,F)) + 2#f(V(g,F))	external sort			1/V(c,F) < 1/f$_s$	E very small
0E	sort(0C)	#fl	3#f(V(g,F))	external sort	output will be l in our case			F very large
				No network, AsterixDB Architecture			Assumption 3L	No network, AsterixDB Architecture
0A	E#$_f$F	#$_f$(|F|/V(c,F))	|E|#$_f$F	Seq Scan		F	w : e.c = f.c	V(C,F) >= V(C,E)
0C	sort(0A)	#$_f$V(g,F)/V(c,F)	|E|(2#f(F/V(c,F)) + 2#f(V(g,F)))	external sort			1/fs < 1/V(c,F)	E very small
0E	sort(0C)	#fl	|E|3#f(V(g,F))	external sort				F very large
				Assumption 3S			No network, AsterixDB Architecture	V(C,F) >= V(C,E)
2A	2#$_f$E	#$_f$E	2#$_f$E	Copy		F	E << M	F >> M
2B	sort(E)	V(c,E)	0	distinct (in memory sort)		T	1/V(c,F) < Mf$_s$/|F|	
2C	join(2B,f,w)	V(c,E)#$_f$F/(max(V(c,E),V(c,F))	V(c,E)|F|/V(c,F)	index join		T		
2E	sort(2C)	V(g,F)/V(c,F)	0	in memory sort				
2G	sort(2E)	#fl	0	in memory sort				
2H	Sort(2G)	2G	0	pre-clustered group by	based on order from 2E			
2I	Join(2H,l.c=r.c)	E#$_f$2G	0	in memory hash join				
				Assumption 3M			No network, AsterixDB Architecture	V(C,F) >= V(C,E)
2A	2#$_f$E	#$_f$E	2#$_f$E	Copy		F	E << M	F >> M
2B	sort(E)	V(c,E)		distinct (in memory sort)		T	1/V(c,F) < 1/f$_s$	
2C	join(2B,f,w)	V(c,E)#$_f$F/(max(V(c,E),V(c,F))	V(c,E)|F|/V(c,F) + #f(V(c,E)F/V(c,F))	index join		F		
2E	sort(2C)	#fV(g,F)V(c,E)	2#f(2C) + 2#fV(g,F)	external sort		F		
2G	sort(2E)	#fl	3#f2E	external sort	Top k result should be small	T		
2H	Sort(2G)	2G	0	pre-clustered group by		T		
2I	Join(2H,l.c=r.c)	E. #$_f$F.s$_w$	0	in memory hash join		T		
				Assumption 3L			No network, AsterixDB Architecture	V(C,F) >= V(C,E)
2A	2#$_f$E	#$_f$E	2#$_f$E	copy		F	E << M	F >> M
2B	sort(E)	V(c,E)	0	distinct (in memory sort)		T	1/fs < 1/V(c,F)	
2C	join(2B,f,w)	V(c,E)#$_f$F/(max(V(c,E),V(c,F))	#$_f$F + #f(V(c,E)F/V(c,F))	hash join	read right from DB	F		
2E	sort(2C)	#fV(g,F)V(c,E)	2#f(2C) + 2#fV(c,E)V(g,F)	external sort		F		

2G	sort(2E)	V(c,E)#fl	3#f2E	external sort	Top k result should be small	T		
2H	Sort(2G)	2G	0	pre-clustered group by		T		
2I	Join(2H,l.c=r.c)	E. $\#_f F.s_w$	0	in memory hash join		T		
				Assumption 3S			No network, AsterixDB Architecture	V(C,F) >= V(C,E)
4A	join(E,F,l.c=r.c)	$\#_f$(EF/max(V(c,E),V(c,F))	E(F/V(c,F))	index join	assume E in memory, but F is not. Output smaller than memory	F	E << M	F >> M
4B	sort(4A)	V(g,F)	0	in memory sort		T	1/V(c,F) < Mf_s/|F|	
4C	sort(4B)	#fl	0	in memory sort		T	l is small	V(g,F) is large
4D	sort(4C)	4C	0	in memory sort		T		
				Assumption 3M			No network, AsterixDB Architecture	V(C,F) >= V(C,E)
4A	join(E,F,l.c=r.c)	$\#_f$(EF/max(V(c,E),V(c,F))	E(F/V(c,F)) + $\#_f$EF/V(c,F)	index join	assume E in memory, but F is not. Output bigger than memory	F	E << M	F >> M
4B	sort(4A)	V(g,F)|E|	2#f(4A) + 2#fV(g,F)|E|	external sort	V(g,F) > M	F	1/V(c,F) < Mf_s/|F|	
4C	sort(4B)	| E | $\#_r$ l	3#f(4C))	external sort		F	l is small	V(g,F) is large
4D	sort(4C)	4C	0	in memory sort		T		
				Assumption 3L			No network, AsterixDB Architecture	V(C,F) >= V(C,E)
4A	join(E,F,l.c=r.c)	$\#_f$(EF/max(V(c,E),V(c,F))	$\#_f$F + #fEF/V(c,F)	hash join	assume E in memory, but F is not. Output bigger than memory		E << M	F >> M
4B	sort(4A)	V(g,F)|E|	2#f(4A) + 2#fV(g,F)|E|	external sort			1/V(c,F) < Mf_s/|F|	
4C	sort(4B)	| E | $\#_r$ l	3#f(4C)	external sort			l is small	V(g,F) is large
4D	sort(4C)	4C	0	in memory sort