Nathan Hernandez
NJH 2349
9/6/2023

ASSIGNMENT 1: Performance Counters

Part 1.)

- **For the smallest matrix size, do the L1 and L2 miss rates vary for the different loop-order variants? Do they vary for the larger matrix sizes? Is there any difference in behavior between the different problem sizes? Can you explain intuitively the reasons for this behavior?**
    - Yes the miss rates vary for the smallest matrix size. With IKJ having the least amount of miss rates and JKI having the most miss rates. The differences with the biggest sizes are still apparent following the same behavior. This behavior is caused by how they access data based on how the loop is set up. This affects the cache locality and it seems like JKI has the worst locality while IKJ has the best.

- **Re-instrument your code by removing PAPI calls, and using clock_gettime with CLOCK_THREAD_CPUTIME_ID to measure the execution times for the six versions of MMM and the eight matrix sizes specified above. How do your timing measurements compare to the execution times you obtained from using PAPI? Repeat this study using CLOCK_REALTIME. Explain your results briefly.**
    - The realtime and the thread time did take longer to run than PAPI. However, the difference between realtime and thread time was not very significant and the differences were due to overhead.

Part 2.)

- **What are *data and control dependences?* Give simple examples to illustrate these concepts.**
    - Data Dependences are when a program statement uses the data of a preceding statement.
        - Raw example:
            - X= 2
            - Y= x + 1
        - War example:
            - Y = x + 1
            - X=2
        - Waw example:
            - X=2
            - X=3
    - Control Dependencies allows the execution of instruction if the previous instruction does not change anything about the current instruction.
        - If (Boolean statement)

        - Instruction;
    - Else
        - Instruction;

- **Explain *out-of-order execution* and *in-order retirement/commit*. Why do high-performance processors execute instructions out of order but retire them in order? What hardware structure(s) are used to implement in-order retirement?**
    - Out of order executes instructions based on their data dependencies and is used to keep the busy waiting down and maximize throughput of the instructions.
    - In order retirement / commit is making sure the instructions complete their execution in the same order they were issued.
    - Processors use out of order to execute instructions to maximize the instruction throughput by working on all available instructions instead of waiting on instructions that are waiting. Processors use in-order retirement to maintain consistency and make sure that interrupts are handled correctly.
    - The in-order retirement is handled in the hardware by the reorder buffer or the ROB. It holds the instructions that completed the execution and waits till instructions reach the head of the ROB to retire.
- **Consider the out-of-order execution model described in lecture (ROB+register renaming). Since there are a limited number of physical registers, the processor must determine when it is safe to reallocate a physical register to hold another value. Explain briefly how a processor might do this.**
    - The ROB checks if the instruction is safe by checking if it's in-flight. If it's in-flight that means the instruction is executing and is not safe to move. Once the execution is completed and moves into retirement the ROB removes the in-flight tag from the instruction. The ROB checks registers to see if any in-flight instruction is depending or writing to it. If the register is free then it renames it to hold another value.

Table and plots for Thread time:

Data is in nanoseconds. Although IJK does very well with cache locality and has low run times I was surprised with the performance of KIJ. JKI did the worst with runtime.

	Thread time							
Sizes	50	100	200	400	800	1200	1600	2000
ijk	152687	1329069	14573050	178482651	5274575872	19146084979	50693955681	120473132849
ikj	294713	1231027	12951403	101718923	855957160	3046046017	7185550011	13836281588
jik	313288	1252709	13428890	175531060	5217126737	19098336184	50299878810	122254571848
jki	382346	2088763	24388938	982959432	10498144016	39112721515	100626410821	209812725617
kij	281627	1252115	12813547	102197820	889807185	4217420326	9726552931	19162942777
kji	376637	2162001	23306344	974907547	10443005325	38539725190	101088017182	211523741889

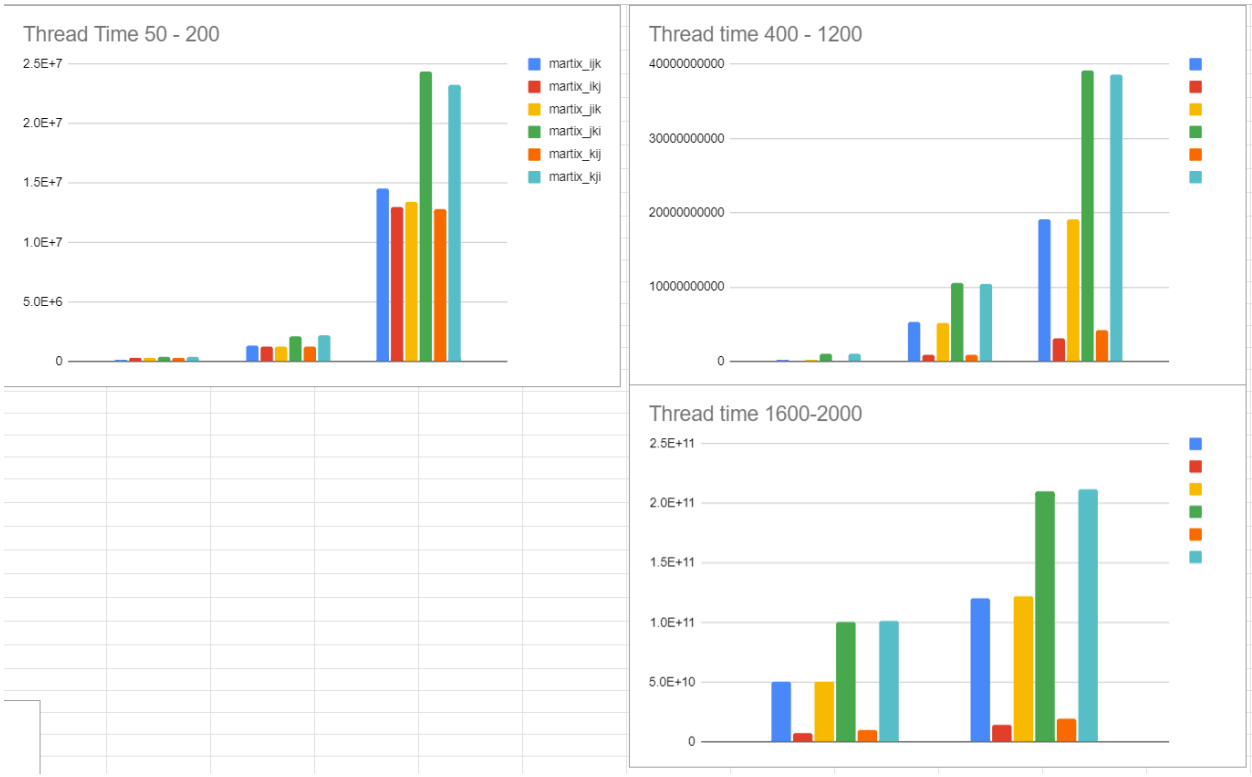


Table and plots for Real time:

Data in nanoseconds. Overall IJK was the fastest with KIJ in second. Similar to thread time the JKI variation did the worst.

	Real time							
Sizes	50	100	200	400	800	1200	1600	2000
martix_ijk	159439	1365271	15280134	181868725	5238377618	18777198268	49801301424	125216366396
martix_ikj	276394	1254367	10464789	103198449	881657435	3128533326	7320940639	14168318336
martix_jik	305711	1333981	13707976	180725497	5291196019	19123650598	49986832916	130476596440
martix_jki	381616	2217718	24141092	987212961	10548987518	38830650726	102037499431	215652711922
martix_kij	286189	1281340	13707863	108752398	944802661	4514293458	10242175509	19786272444
martix_kji	385448	2097255	23774404	988226155	10585329644	39748477808	102556334233	213142524333

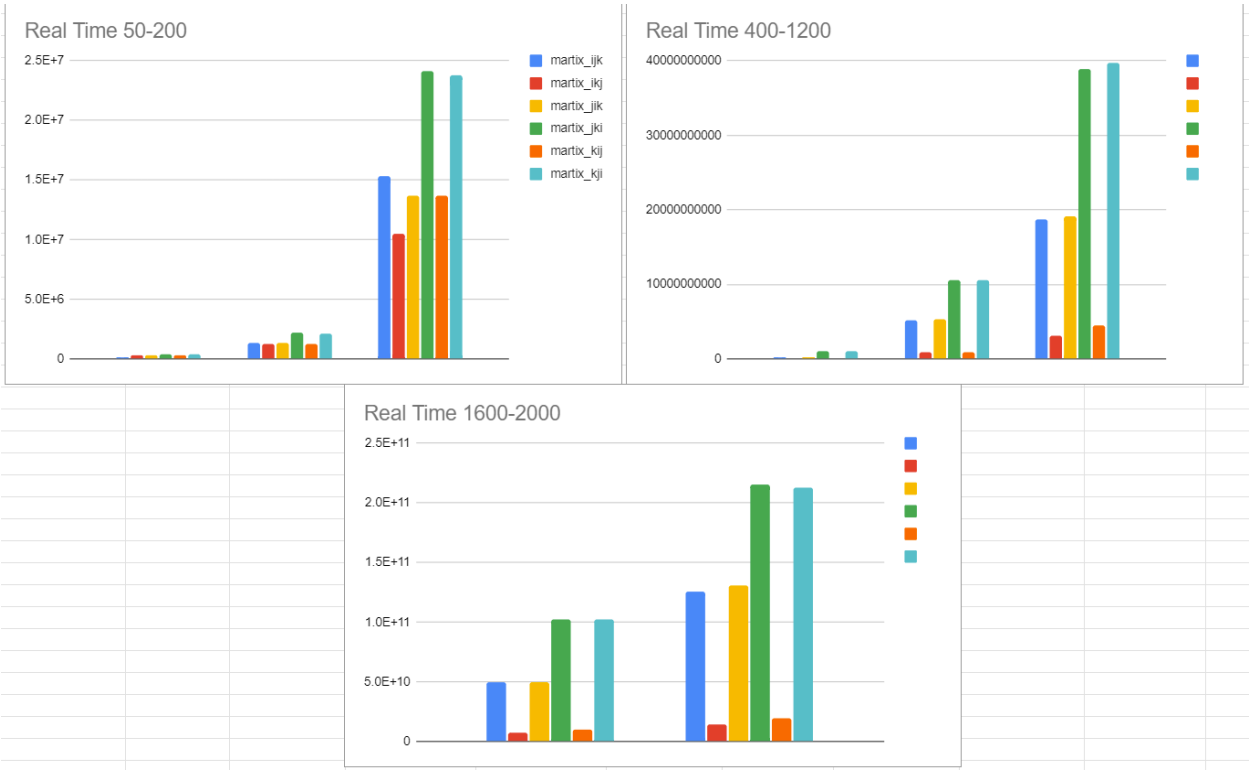


Table and plots for Papi:

The IKJ variation stands above the rest as being the fastest and with less Cache Misses in L1 and L2. As well as less Cycles than other variants.

Var: IJK

Size	ijk	PAPI_LD_INS	PAPI_SR_INS	PAPI_FP_INS	PAPI_L1_DCM	PAPI_L2_DCM
50		382828	125162	250216	7051	502
100		3030541	1000201	2007193	199036	22766
200		24120224	8000164	16009138	1056941	796339
400		192480267	64000139	128204046	64790392	8174174
800		1537922789	512002725	1027907013	944926840	68606353
1200		5188333753	1728013689	3464149284	3695797257	428581286
1600		12295701357	4096021293	8206259601	9246234948	2688457695
2000		24012046647	8000046581	16018660983	18081416782	10459765859

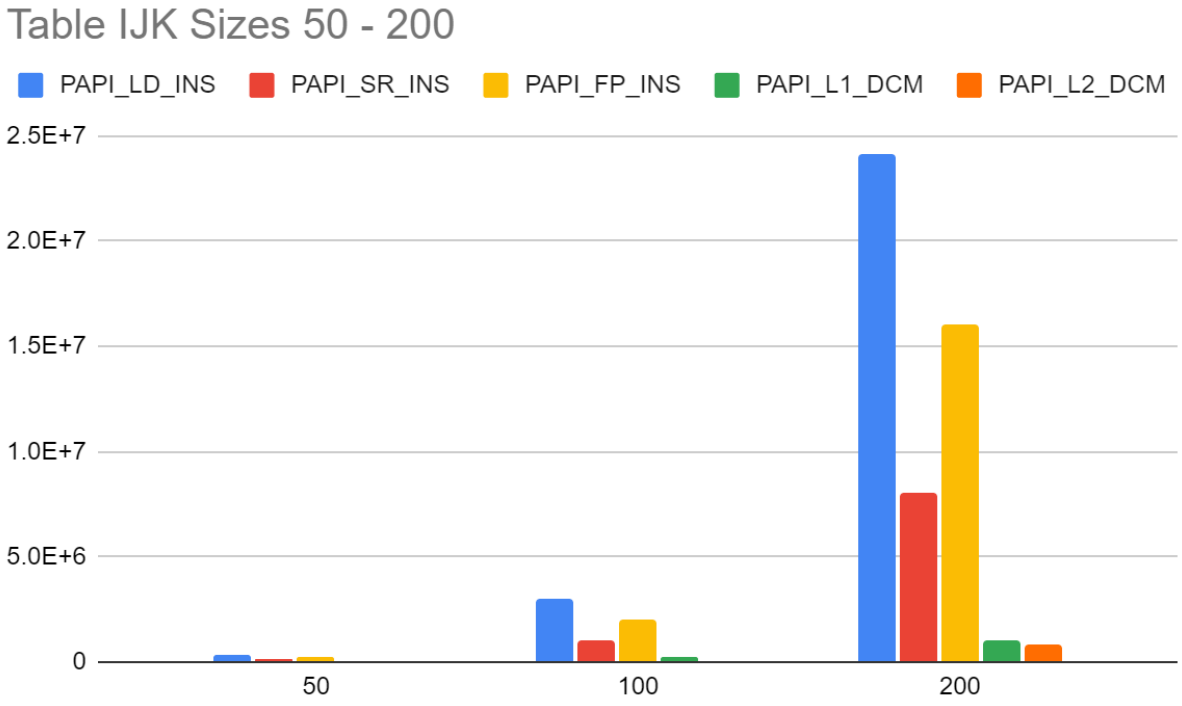


Table IJK Sizes 50 - 200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
2.5E+7
2.0E+7
1.5E+7
1.0E+7
5.0E+6
0
50
100
200

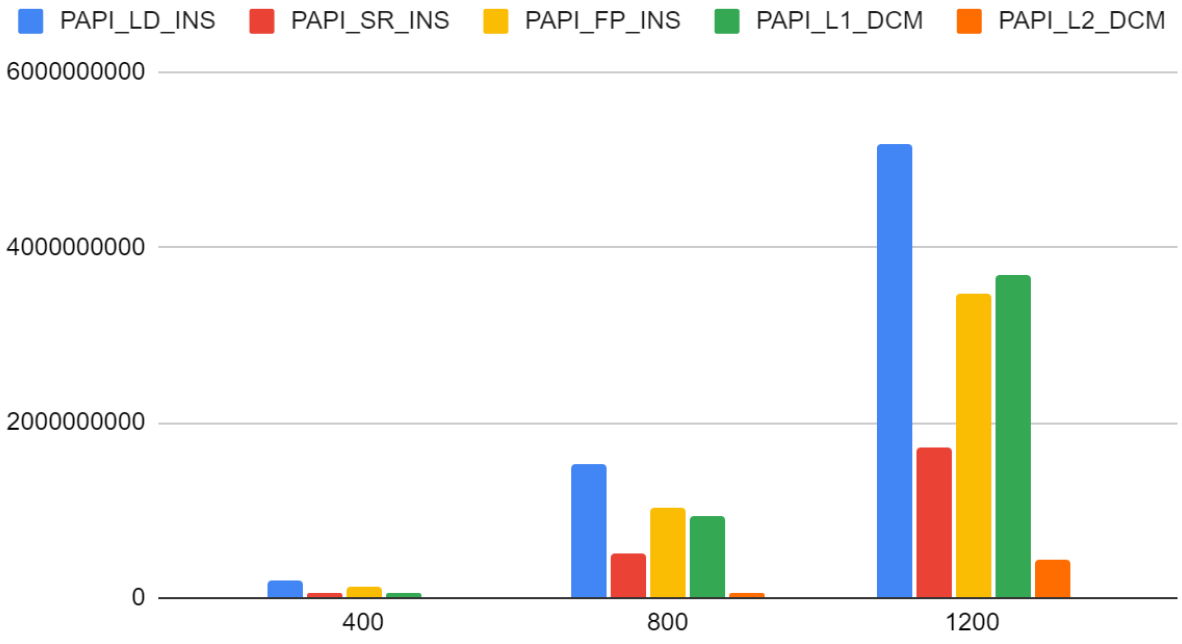
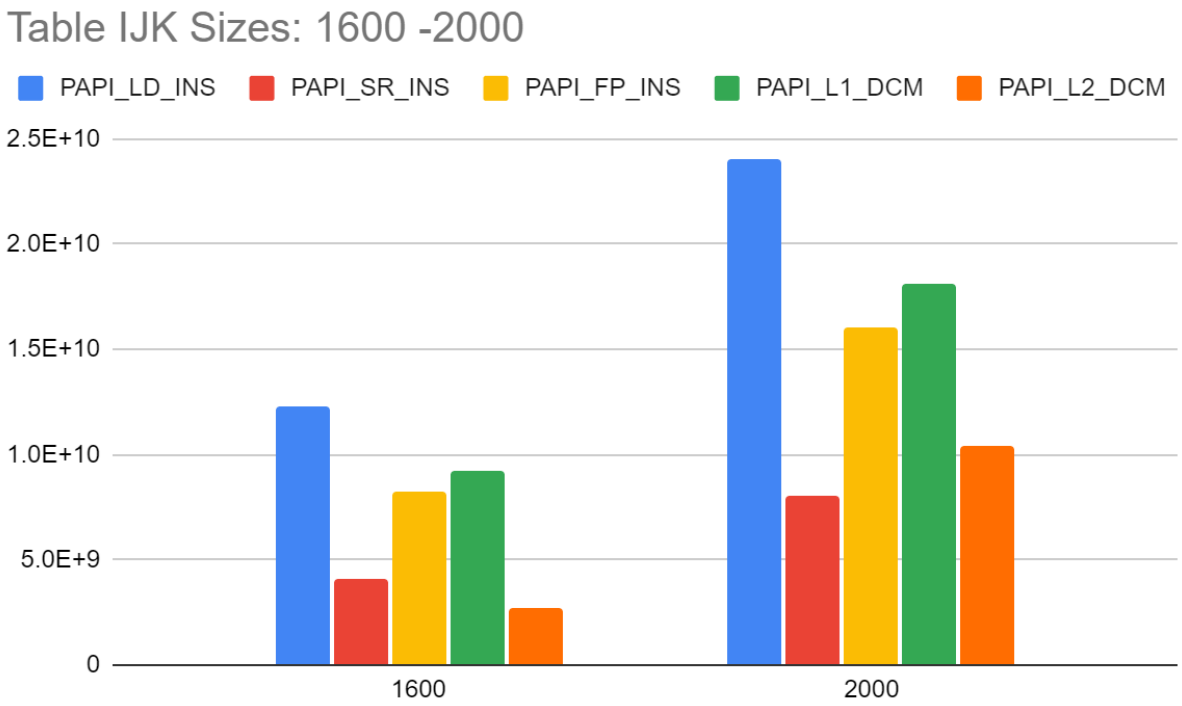


Table IJK Sizes: 400 - 1200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
6000000000
4000000000
2000000000
0
400
800
1200



Var: IKJ

Size	ikj	PAPI_LD_INS	PAPI_SR_INS	PAPI_FP_INS	PAPI_L1_DCM	PAPI_L2_DCM
50		382641	125077	268200	2617	402
100		3030141	1000109	2006217	141891	1433
200		24120145	8000115	16002453	1078903	106591
400		192480170	64000119	128003193	8396437	491692
800		1537921279	512001224	1024252510	65900012	2630021
1200		5188324266	1728004208	3458544760	221925083	7915320
1600		12295689025	4096008962	8197061645	529999032	17916389
2000		24012015935	8000015872	16008946302	1225225182	33629302

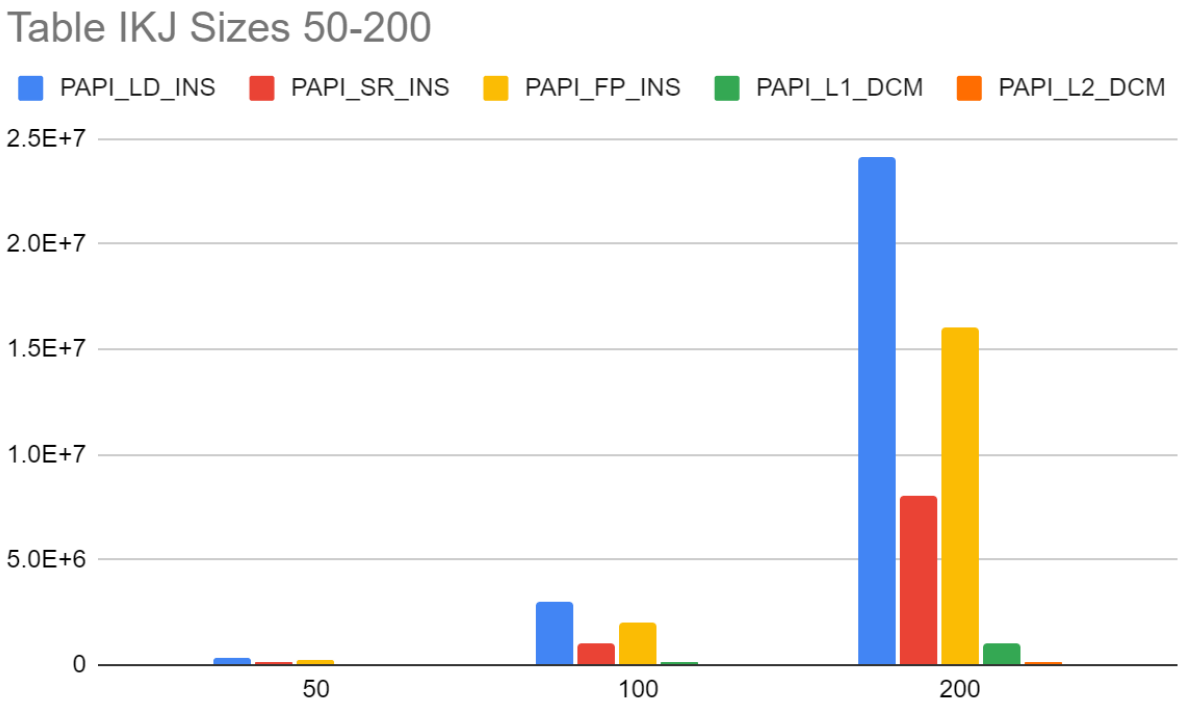


Table IKJ Sizes 50-200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
2.5E+7
2.0E+7
1.5E+7
1.0E+7
5.0E+6
0
50
100
200

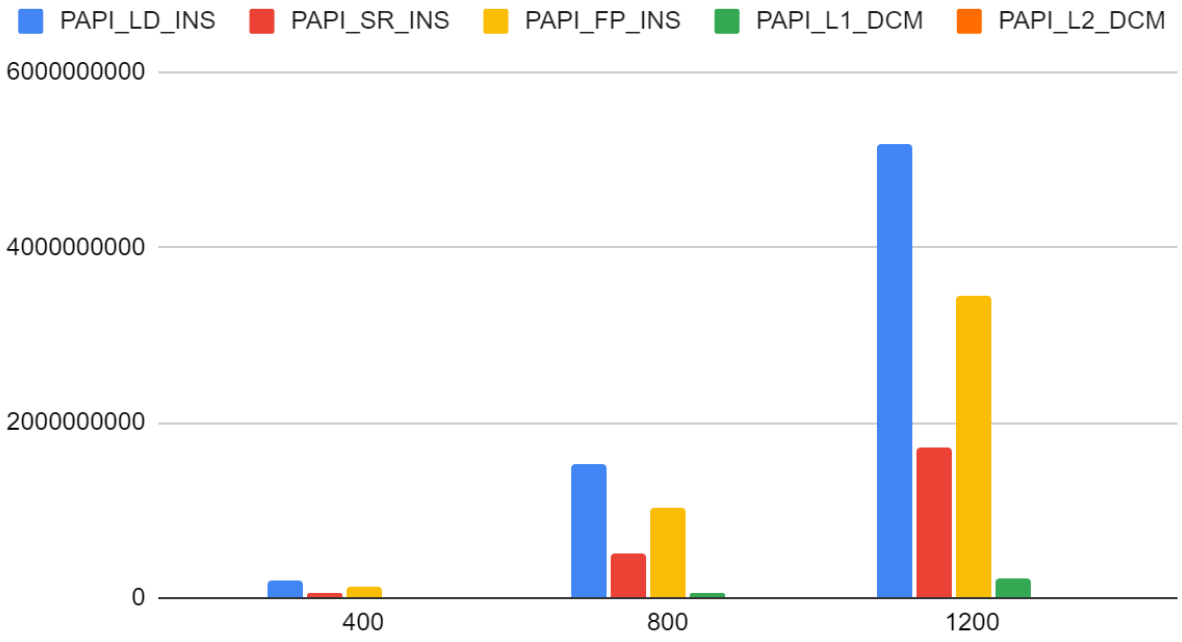
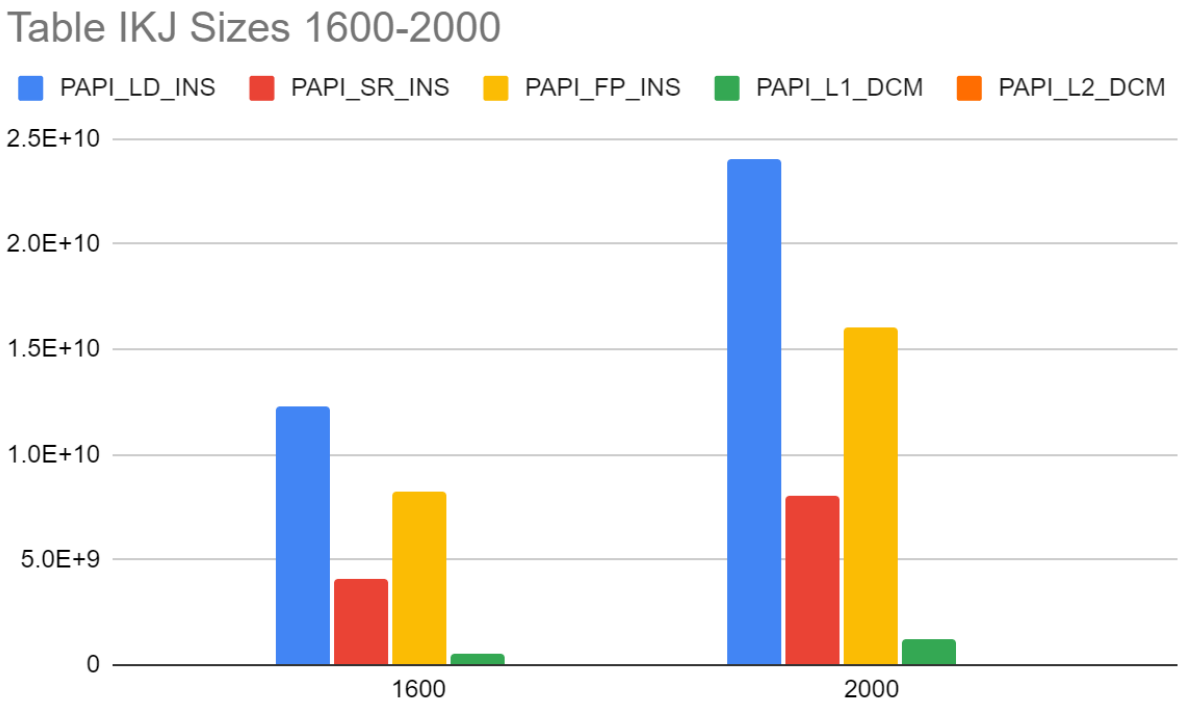


Table IKJ Sizes 400-1200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
6000000000
4000000000
2000000000
0
400
800
1200



Var: JIK

Size	jik	PAPI_LD_INS	PAPI_SR_INS	PAPI_FP_INS	PAPI_L1_DCM	PAPI_L2_DCM
50		384053	125388	270085	8541	410
100		3030625	1000127	2006184	147262	5180
200		24120340	8000080	16006998	1141481	200724
400		192480281	64000131	128197903	66343468	1135856
800		1537922613	512002523	1027608711	945217895	13350040
1200		5188329644	1728009567	3463111029	3696307217	240711255
1600		12295700879	4096020811	8204523382	9247596865	2453959727
2000		24012046494	8000046426	16017340534	18082415145	10145528507

Table JIK Sizes 50- 200

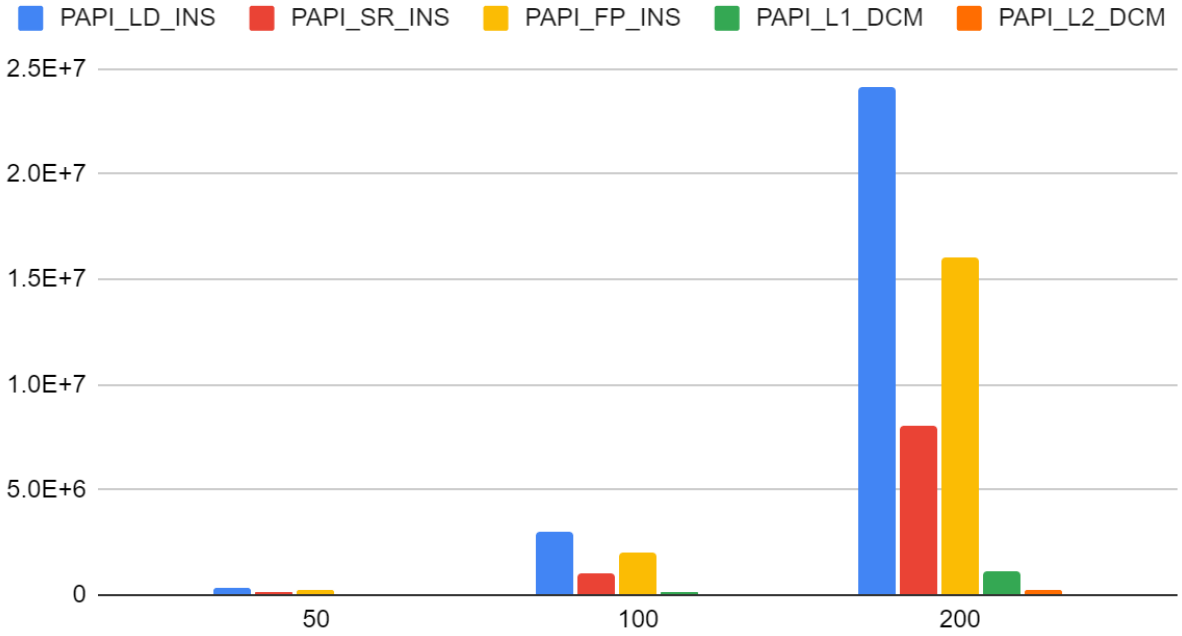
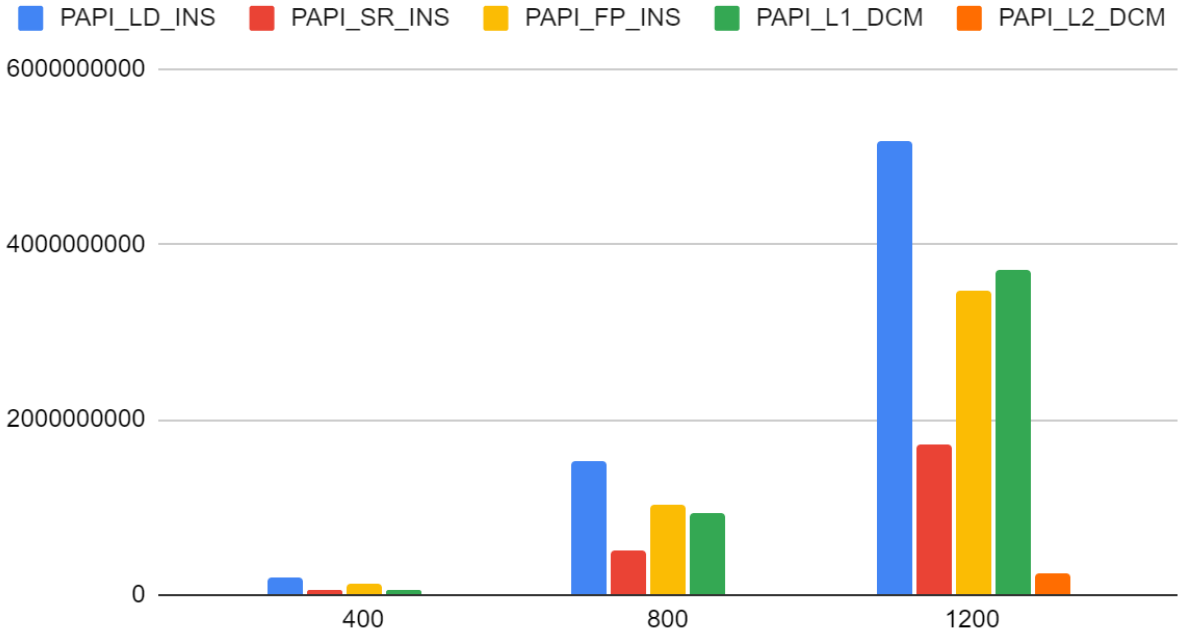
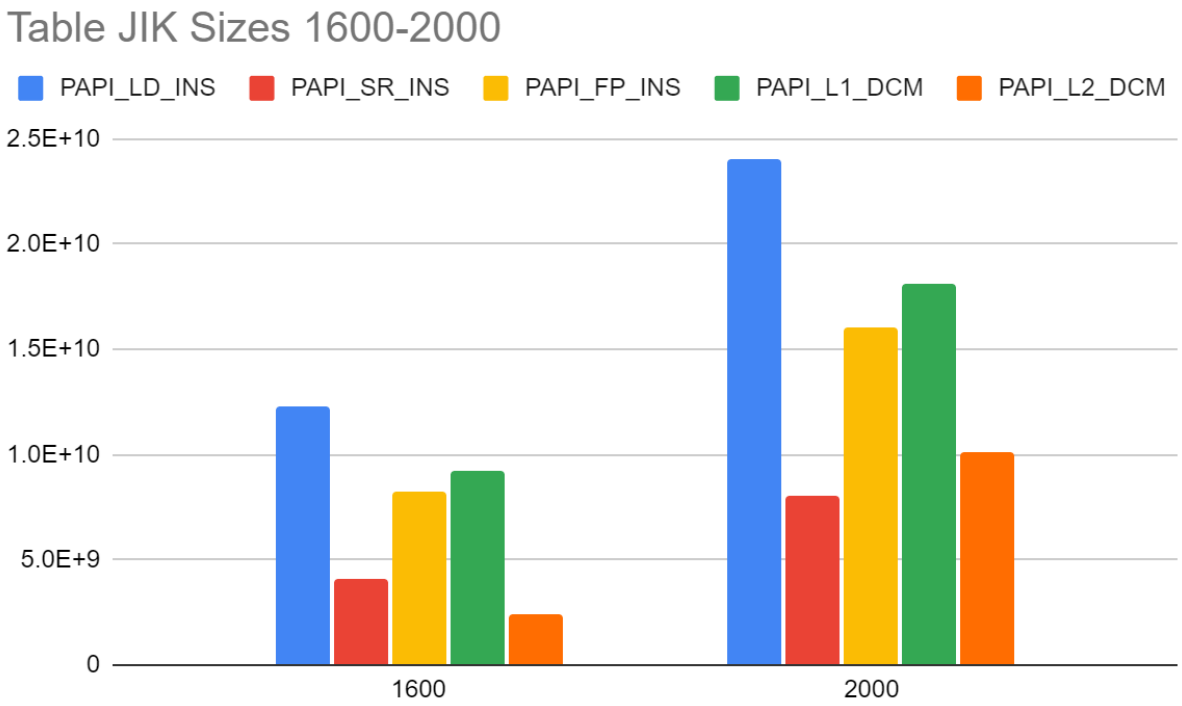


Table JIK Sizes 400-1200





Var:JKI

Size	jki	PAPI_LD_INS	PAPI_SR_INS	PAPI_FP_INS	PAPI_L1_DCM	PAPI_L2_DCM
50		630140	125093	261553	14684	511
100		5010141	1000114	2010412	161422	3903
200		40040147	8000133	16040842	2682453	981355
400		320160417	64000413	128120605	163570476	8584851
800		2560643929	512003892	1024529492	1458346706	107162218
1200		8641454481	1728014433	3459760837	5427900212	1682681623
1600		20482596620	4096036569	8197501619	14028604701	7845594262
2000		40004072497	8000072441	16016546254	29310041183	19713742757

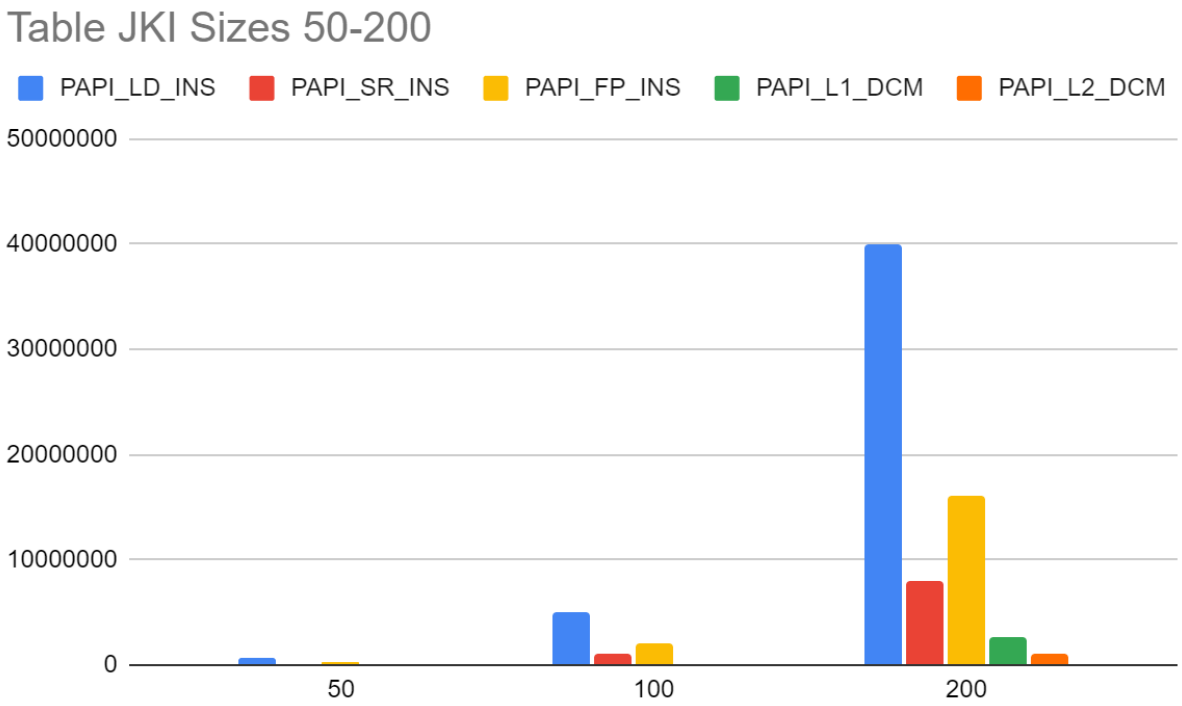


Table JKI Sizes 50-200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
50000000
40000000
30000000
20000000
10000000
0
50
100
200

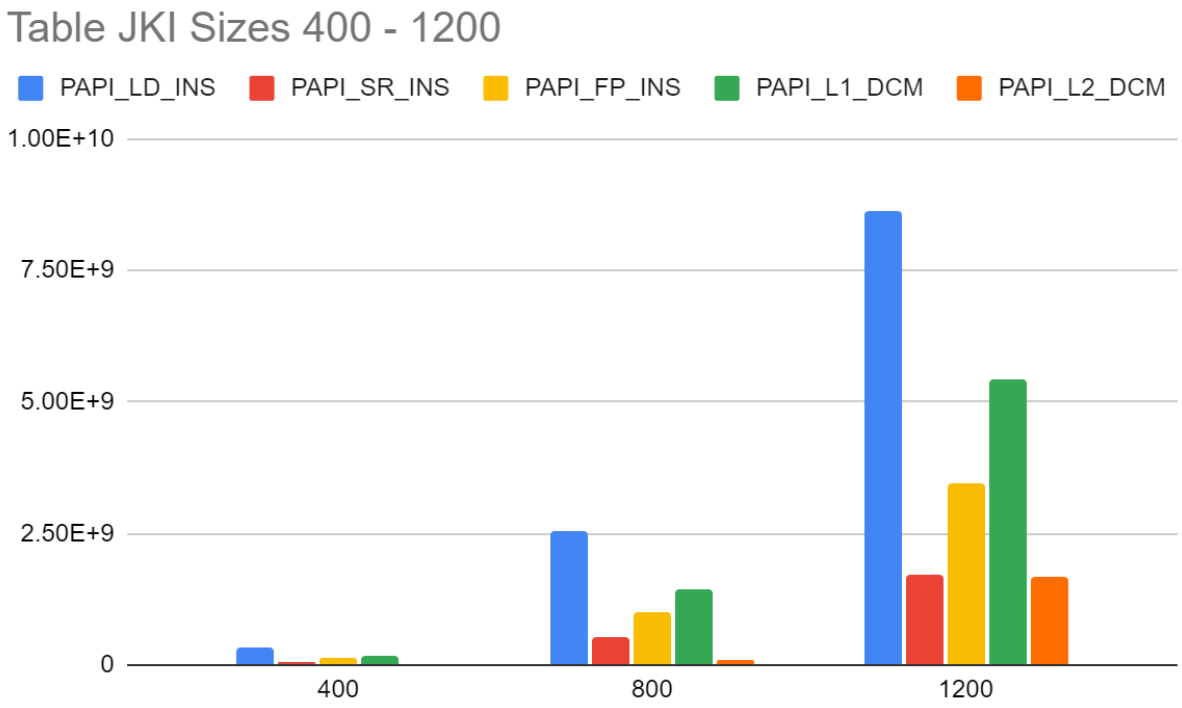
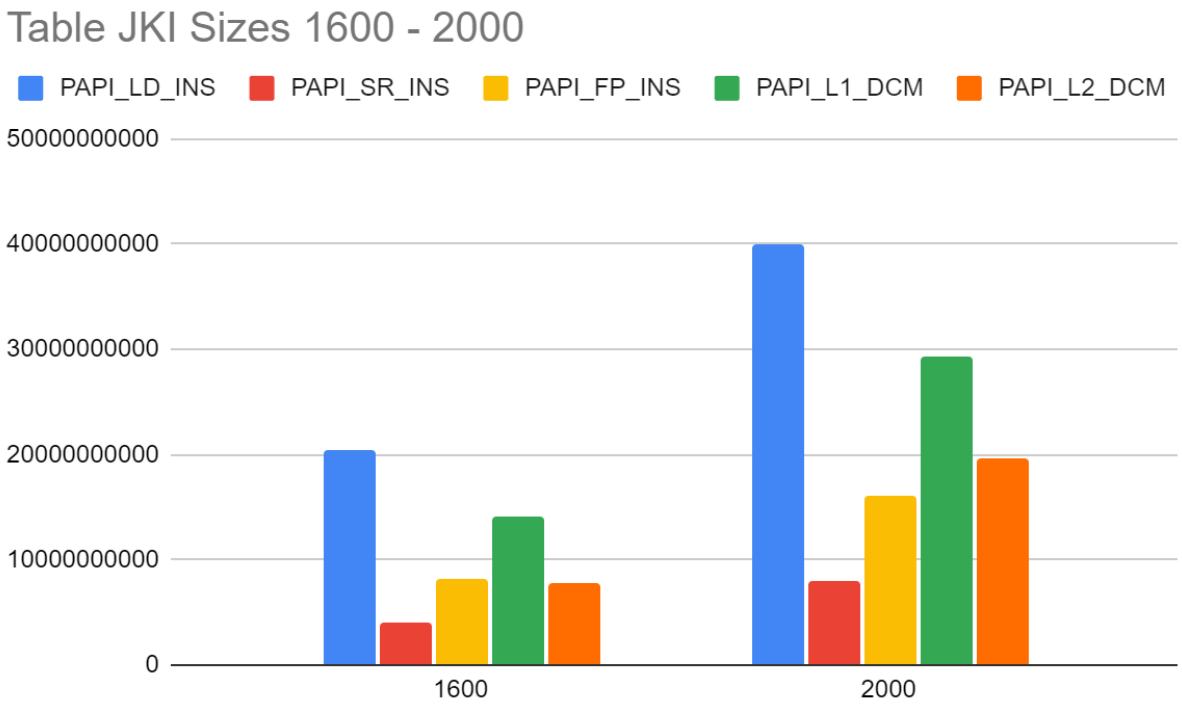


Table JKI Sizes 400 - 1200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
1.00E+10
7.50E+9
5.00E+9
2.50E+9
0
400
800
1200



Var: KIJ

Size	kij	PAPI_LD_INS	PAPI_SR_INS	PAPI_FP_INS	PAPI_L1_DCM	PAPI_L2_DCM
50		382641	125078	268689	7535	432
100		3030141	1000107	2006581	160712	1954
200		24120145	8000138	16002960	1156569	337329
400		192480169	64000120	128007422	8712016	964207
800		1537921282	512001226	1024523799	67159255	4401425
1200		5188324518	1728004458	3459564212	224461208	24636464
1600		12295689880	4096009816	8201732262	536444123	70008538
2000		24012017660	8000017596	16019076293	1258260487	105690326

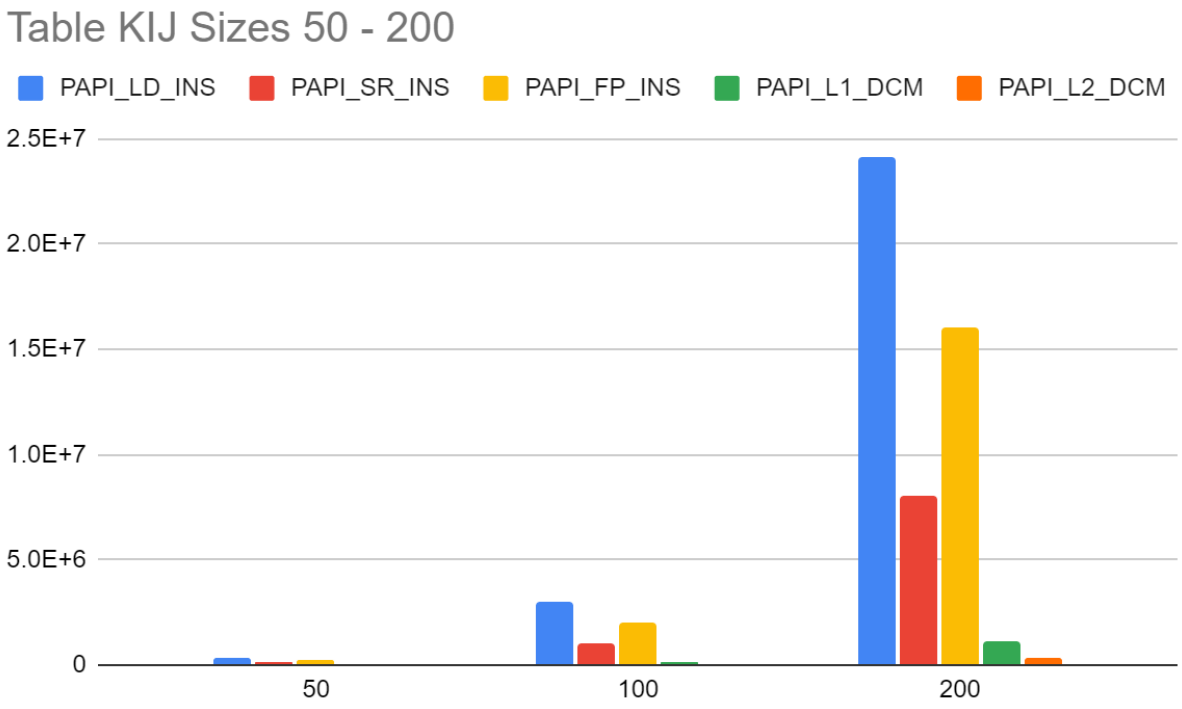


Table KIJ Sizes 50 - 200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
2.5E+7
2.0E+7
1.5E+7
1.0E+7
5.0E+6
0
50
100
200

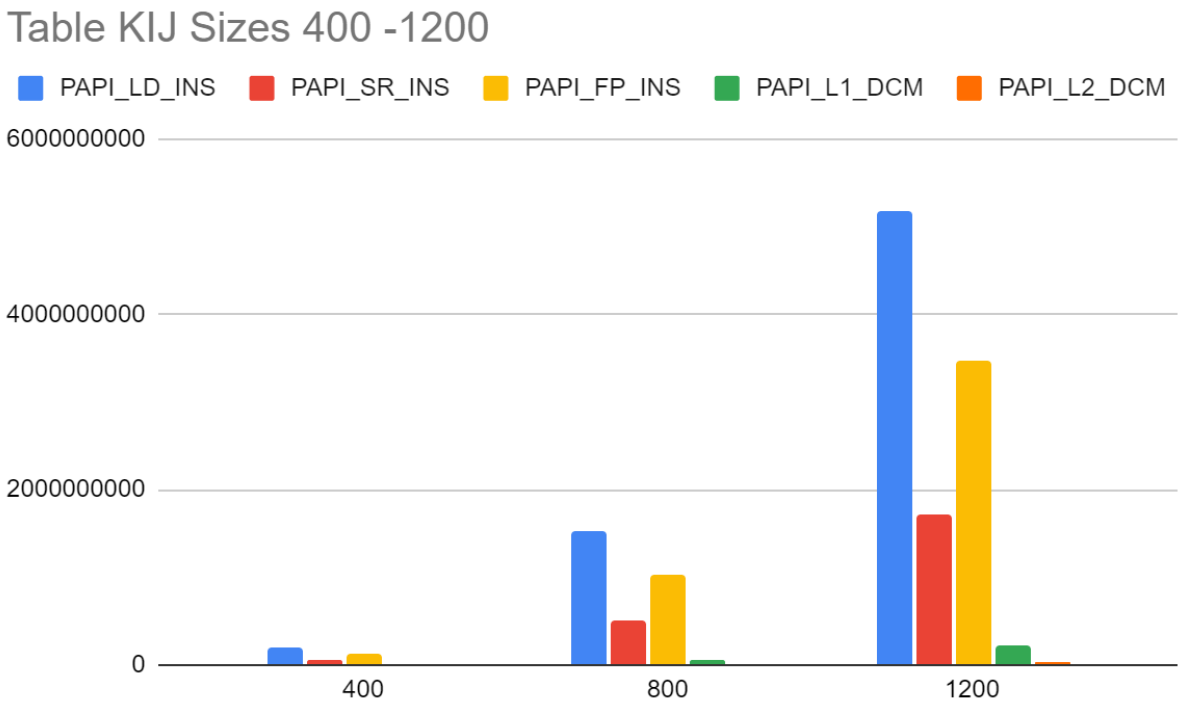
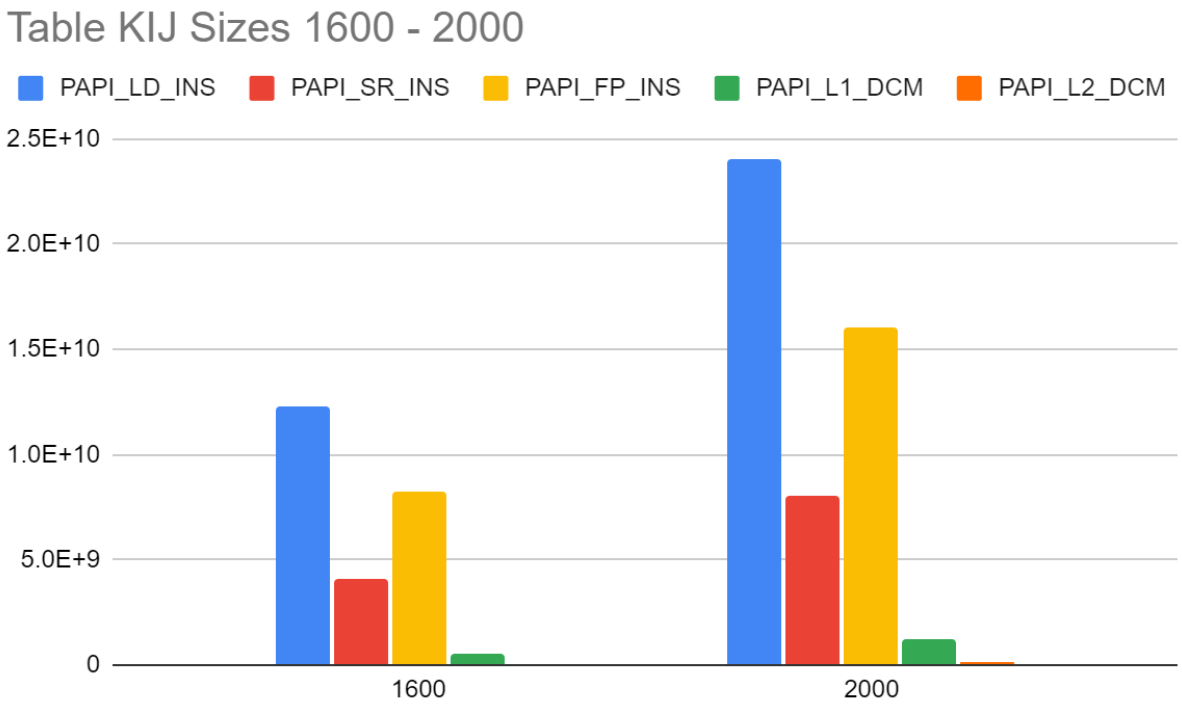


Table KIJ Sizes 400 -1200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
6000000000
4000000000
2000000000
0
400
800
1200



Var: KJI

Size	kji	PAPI_LD_INS	PAPI_SR_INS	PAPI_FP_INS	PAPI_L1_DCM	PAPI_L2_DCM
50		630140	125105	262298	8883	506
100		5010140	1000089	2013171	139677	2518
200		40040147	8000105	16062329	2559565	833701
400		320160443	64000433	128136254	163325946	8294093
800		2560644123	512004081	1025029531	1456810409	117892750
1200		8641454442	1728014393	3462253132	5424567466	1626022119
1600		20482596298	4096036241	8206114541	14021866900	8015074374
2000		40004073671	8000073613	16031922162	29299125722	19795588100

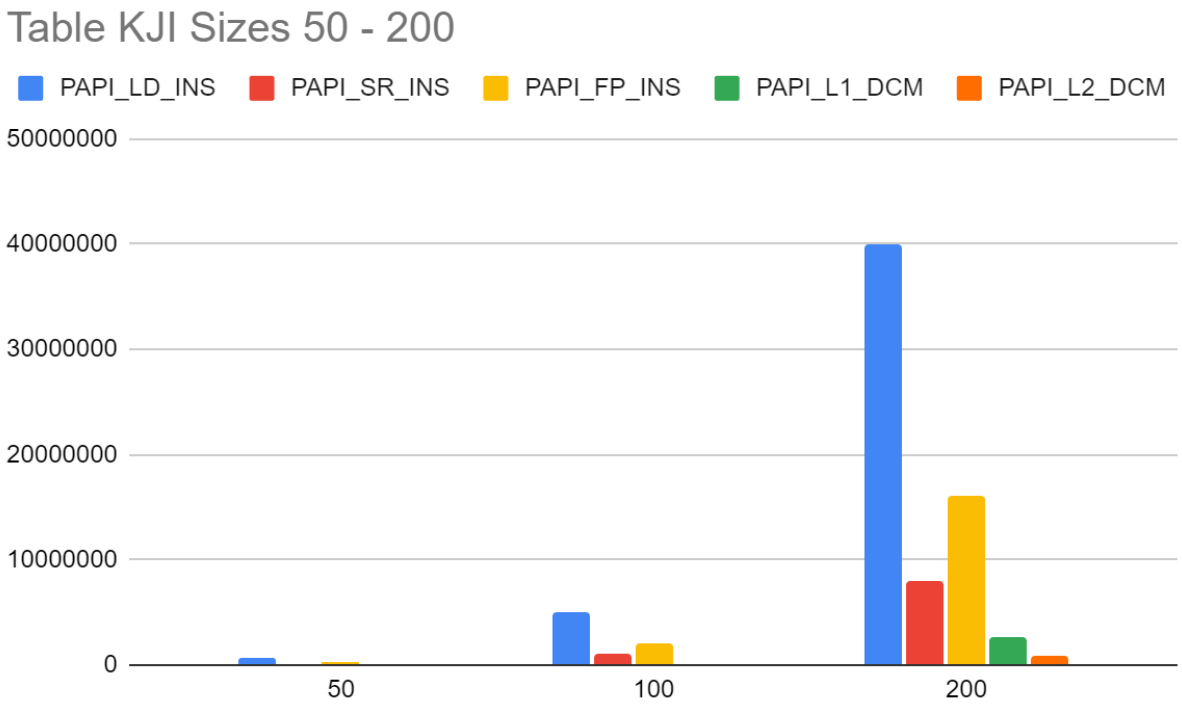


Table KJI Sizes 50 - 200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
50000000
40000000
30000000
20000000
10000000
0
50
100
200

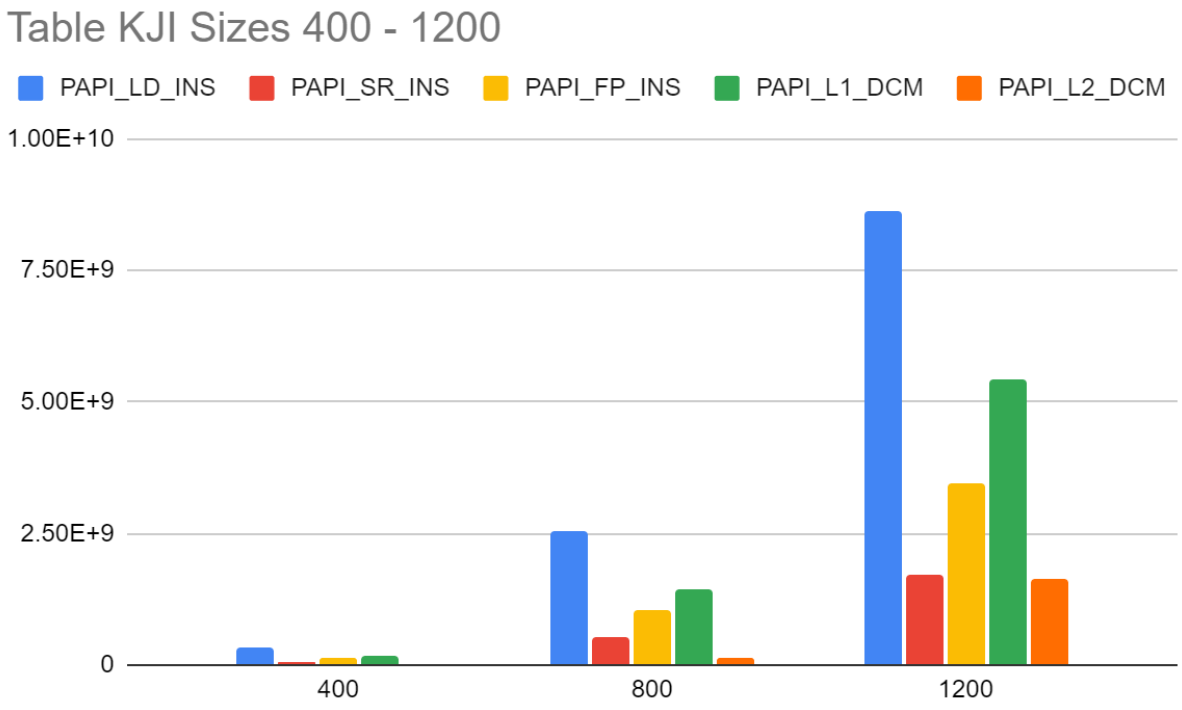


Table KJI Sizes 400 - 1200
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
1.00E+10
7.50E+9
5.00E+9
2.50E+9
0
400
800
1200

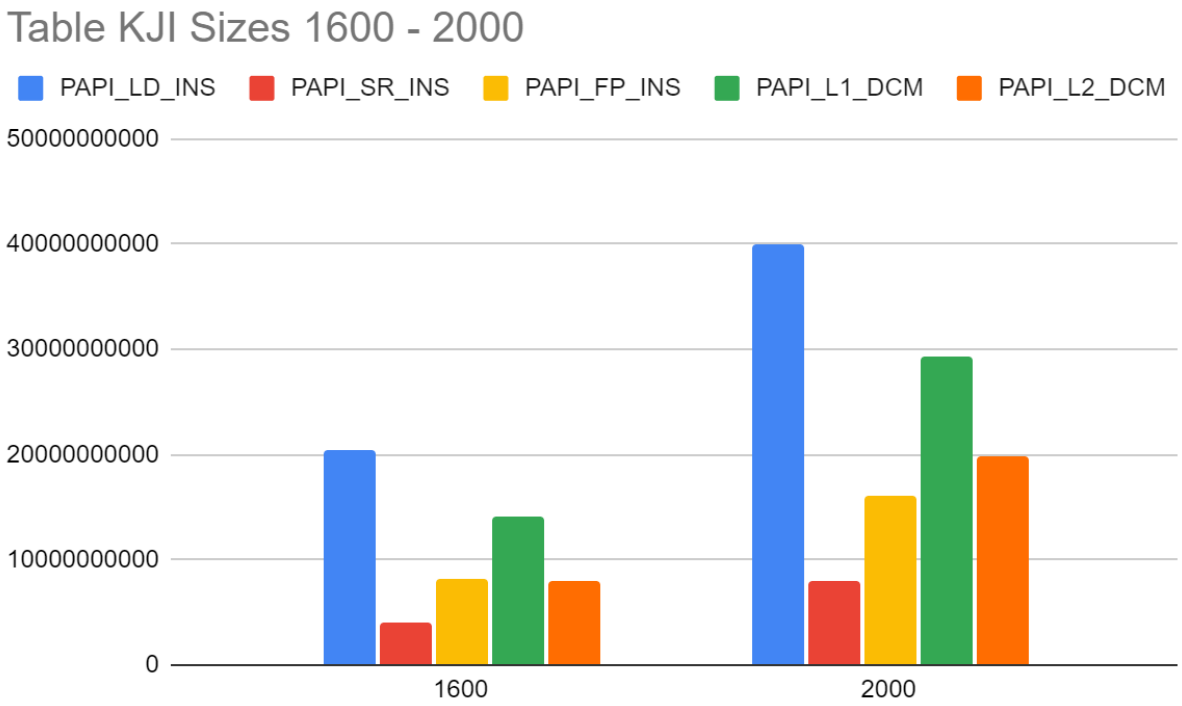


Table KJI Sizes 1600 - 2000
PAPI_LD_INS
PAPI_SR_INS
PAPI_FP_INS
PAPI_L1_DCM
PAPI_L2_DCM
50000000000
40000000000
30000000000
20000000000
10000000000
0
1600
2000