Linear algebra problems with APL2
Performance comparison on different IBM platforms

Renzo Beltrame[1]

In the past years our computer service migrated from the main-frame IBM to the IBM SP1. A further migration from SP1 to SP2 was made in the spring 1995.

We present here a brief comparison of the performance of the three IBM systems, main-frame, SP1, and SP2. Some data are added, which refer to enhanced nodes of SP2, and to two different nodes type and configuration.

The comparison was made to offer to our users a first idea of the performance they can expect after the migration on the new architecture. Therefore the reported data concern a multiusers environment with a heavy load on each node, which is the common operational situation of our computer service.

We choose for the comparison three problems of the linear algebra: the matrix product, the solution of a system of linear equations, and the calculation of eigenvalues and eigenvectors of a square matrix.

The comparison was made by using the APL2 functions and operators that are implemented to solve linear algebra problems, because they are implemented as primitive or plugged functions.

APL2 has usually a good implementation of the linear algebra. Furthermore it is a good candidate for quickly prototyping the solution of many numerical problems. Therefore the test has a reasonable interest.

Nevertheless we did not repeat the test to evaluate the average value of the times and the variance of the distribution sample, because we are not doing a real performance evaluation. The small times we report here for completeness sake, have thus a low significance.

Hardware and Software configuration

The hardware and software configurations were the following ones.

The main-frame was an IBM ES / 9000 9121 / 440, with 2 processors, and vector feature.

The shared memory for the VM / ESA is 48 MB; and the address space of each processor is 2 GB.

We used APL2 2.2.00.

The data concerning the main-frame and SP1 firstly appeared in November 1994 on the CNUCE Report C94-32; the first set of data on SP2 appeared on the CNUCE Report C95-54.

The SP1 was an IBM 9076-102 (SP1), 8 nodes at 66 MHz Thin.
Every node has 64 MB of RAM, and 1 GB hard-disk.
We used APL2 / 6000 Ver. 1.2.0, and LAPACK Version 1.1.
The compiler was the IBM XL Fortran for AIX 3.2.

The SP2 was an IBM 9076-302 (SP2), 8 nodes at 67 MHz Thin.
Every node has 64 MB of RAM, and 2 GB hard-disk.
We used APL2 / 6000 Ver. 1.2.0, LAPACK Version 2.0, and ScaLAPACK Version 1.1.
The compiler was the IBM XL Fortran for AIX 3.2.

The enhanced node of the SP2 (node 1 , and 2) have the following configuration:
IBM 9076-302 (SP2) 67 MHz Thin2.
Every node has 512MB of RAM, and 2 GB hard-disk.
We used APL2 / 6000 Ver. 1.2.0.
The SP2 performance on these nodes (single node programs only) is reported below with the indication "enhanced node".
The UNIX operating system on SP1 and SP2 was in all the cases the AIX 3.2.5.

The data labelled as SP2-2 refer to a node of an IBM 9076 SP with the following configuration:
Processor: power 2 (Thin2sc 120MH)
256 MB Ram (4x64 MB)
4.5 GB internal hard-disk
18.2 GB external hard-disk
Adapter SCSI 2 Differ. Fast Wide (connected to a digital TL812)
Adapter SCSI
Adapter ATM (Turboways 155)
Adapter SSA 4 port raid adapter (connected to 2 hard-disks of the unit 7133 non raid)
Adapter ethernet
Operating sistem AIX 4.1.5

The data labelled as SP2-2 refer to a node of an IBM 9076 SP with the following configuration:
Processor: 8 processors power pc (604 High 112MH?)
1024 MB Ram (4x256 MB)
4.5 GB internal hard-disks (2x2.2 GB)
27 GB external hard-disk
Adapter SCSI 2 Differ. Fast Wide
Adapter SCSI 2 single ended
Adapter SCSI 2 Fast Wide
Adapter ATM (Turboways 155)
Adapter SSA 4 port raid adapter (connected to 4 hard-disks of the unit 7133 raid 5)
Adapter ethernet
Operating sistem AIX 4.1.5

The data labelled as SPC refer to an IBM 7006 40P (SPC) with the following configuration:
Processor: Power PC 601

64 MB Ram (4x16 MB)
4.5 GB internal hard-disks (2x2.2 GB)
Adapter SCSI
Adapter ethernet
Operating sistem AIX 4.1.5

Still in the last three cases we used APL2/6000 Ver. 1.2.0.

APL2 Test

For these tests we generate rectangular matrices of random numbers by using the primitive APL2 function ‘roll’, which generate a random integer in the range of its argument. Iterated calls of this function generate a pseudo random sequence of numbers.

The numbers were chosen in the range of 1 to the product *nm* of the matrix rows and columns, then they were divided by 0.1000000001 to obtain floating point numbers. In APL2 floating point arithmetic is double precision.

We recorded the computer times by using the suitable APL2 system function, and the times refer only to the execution of the linear algebra operation. All the times are given in milliseconds.

Product of two matrices

The first set of data (see Tables 1-7) concerns the product of two matrices, which have *n* rows, and *m* columns. In APL2 we realize this operation by means of primitive functions and operators. In particular, we assigned to a variable the transpose of the second matrix, and the time is so excluded of doing the transpose operation.

In the Tables *n* labels the rows, and *m* labels the columns; both matrices were prepared with the suitable number of rows and columns. The time of the operation is given in milliseconds.

In Table 1 we have the data obtained on the main-frame IBM when the vector feature was disabled. As we expect, they are very poor.

	50	100	150	200	250	300	350	400	450
50	31	66	119	257	363	459	562	664	757
100	132	292	453	636	833	1040	1267	1511	1804
150	305	638	973	1283	1615	1987	2413	3340	4954
200	568	1114	1648	2195	2811	3584	4843	6230	7703
250	887	1724	2574	3479	4611	5985	7508	9358	13170
300	1279	2479	3749	5106	6692	8471	10138	13913	18898
350	1731	3376	5157	7071	9222	11555	15497	21601	30531
400	2258	4398	6763	9216	11624	14974	20848	31120	44399
450	2841	5600	8561	11737	15353	20911	33527	48911	63549

Table 1: APL2 on VM-CMS without vector feature. Product of two rectangular matrices

The matrix of the times is highly not symmetric, because the elements, which belong to either the same row or the same column, are stored in contiguous locations of memory. Therefore architectural facts strongly influence the performance, like strategies of memory access, piping, caching, buffering, etc. In our case the product of two matrices 50x450 requires 3.753 times the time required by two 450x50 matrices.

In Table 2 we have the data obtained on the main-frame IBM when the vector feature was enabled. As we can see, we have a good performance.

Here too the matrix of the performance times is highly not symmetric; and the product of two matrices 50x450 requires 6.8 times the time required by two 450x50 matrices. This figure reduces to 1.12 among 400x450 and 450x400 matrices.

	50	100	150	200	250	300	350	400	450
50	4	8	14	27	35	43	48	55	62
100	12	44	67	88	110	135	156	177	200
150	31	93	139	186	239	282	328	371	424
200	80	159	236	316	396	478	555	632	707
250	124	244	363	489	610	737	854	972	1085
300	187	383	583	777	977	1166	1354	1550	1756
350	256	520	770	1031	1286	1538	1807	2080	2331
400	322	648	975	1310	1630	1967	2291	2621	2992
450	422	836	1260	1681	2082	2477	2932	3375	3787

Table 2: APL2 on VM-CMS with vector feature. Product of two rectangular matrices

In Table 3 we have the data obtained on the SP1, and the performance is quite similar to the mainframe performance when the vector feature is enabled.

Here again the matrix of the performance times is higly not symmetric; and the product of two matrices 50x450 requires 10.7 times the time required by two 450x50 matrices. This figure reduces to 1.135 among 400x450 and 450x400 matrices.

	50	100	150	200	250	300	350	400	450
50	6	17	23	30	40	48	58	64	72
100	32	62	93	121	150	180	215	250	290
150	70	140	206	284	350	430	480	556	630
200	130	250	380	504	630	746	860	980	1100
250	215	410	590	782	980	1141	1340	1530	1690
300	320	580	850	1130	1380	1650	1921	2180	2470
350	430	800	1130	1550	1930	2280	2650	3010	3400
400	602	1074	1550	2050	2520	3000	3480	3960	4430
450	770	1370	1970	2600	3190	3790	4420	5030	5720

Table 3: APL2 on SP1. Product of two rectangular matrices

In Table 4 we have the data obtained on the SP2, and the performance is quite similar to the mainframe performance when the vector feature is enabled.

Here again the matrix of the times is highly not symmetric; and the product of two matrices 50x450 requires 8.23 times the time required by two 450x50 matrices. This figure reduces to 1.15 among 400x450 and 450x400 matrices.

	50	100	150	200	250	300	350	400	450
50	4	10	17	31	30	43	50	60	68
100	20	60	80	110	140	170	210	220	260
150	60	150	220	260	320	380	430	480	530
200	130	210	330	430	530	660	750	860	980
250	170	350	510	660	870	990	1180	1300	1500
300	270	480	710	920	1210	1440	1690	1940	2180
350	370	680	980	1300	1640	1990	2510	2610	2980
400	490	890	1340	1790	2190	2610	3020	3440	3790
450	560	1190	1730	2360	2830	3470	3830	4360	4990

Table 4: APL2 on SP2. Product of two rectangular matrices

In Table 5 we report the data of the enhanced node of the SP2.

As we expect the performance is better than the previous one, particularly when the matrices dimensions increase. Here again the matrix of the times is highly not symmetric. the product of two matrices 50x450 requires 9.4 times the time required by two 450x50 matrices. This figure reduces to 1.146 among 400x450 and 450x400 matrices.

	50	100	150	200	250	300	350	400	450
50	4	7	12	19	25	30	38	45	50
100	17	34	64	80	120	120	150	170	210
150	30	80	140	190	230	290	320	370	420
200	60	190	260	340	420	480	580	640	780
250	130	240	400	500	630	770	910	1010	1090
300	180	340	560	710	920	1060	1300	1480	1680
350	280	530	750	960	1220	1510	1770	2100	2360
400	330	670	1060	1340	1650	1970	2370	2620	2950
450	470	830	1340	1660	2160	2530	2900	3380	3780

Table 5: APL2 on SP2. Product of two rectangular matrices (enhanced node)

In Tables 6-7 we have the data obtained on two different nodes of the SP2.

The Table 6 refers to the node labelled SP2-2, and the performance is the best in the full test. Here again the matrix of the times is highly not symmetric; and the product of two matrices 50x450 requires 8.93 times the time required by two 450x50 matrices. This figure reduces to 1.1 among 400x450 and 450x400 matrices. Hence we reach the best performance still from this point of view.

The Table 7 refers to the node labelled SP2-5, and the performance is worse than the performance we obtained on the SP1. Here again the matrix of the times is highly not symmetric; and the product of two matrices 50x450 requires 8.74 times the time required by two 450x50 matrices. This figure reduces to 1.112 among 400x450 and 450x400 matrices. From this point of view the performance is qualitatively similar to previous one.

	50	100	150	200	250	300	350	400	450
50	1	4	7	11	16	20	24	26	30
100	9	22	36	49	61	64	85	95	107
150	22	53	81	106	132	157	182	211	237
200	40	94	140	185	233	278	326	371	419
250	75	147	217	289	361	435	510	573	647
300	111	214	317	418	522	620	723	830	924
350	156	299	431	553	690	835	970	1106	1248
400	210	386	562	743	909	1101	1251	1442	1638
450	268	489	710	932	1150	1371	1583	1802	2055

Table 6: APL2 on SP2. Product of two rectangular matrices (SP2-2)

	50	100	150	200	250	300	350	400	450
50	10	20	31	46	61	72	78	102	108
100	44	94	148	214	266	316	371	442	521
150	92	204	326	448	614	745	891	1002	1209
200	179	397	628	804	1106	1214	1685	2087	2207
250	279	638	1022	1316	1779	2357	2854	3233	3828
300	376	899	1451	2019	2790	3498	4110	4991	5568
350	579	1367	1995	2992	3961	4650	5914	6799	7886
400	807	1710	2837	3961	5158	6630	7975	9232	10682
450	944	2253	3721	5182	6809	8505	10140	11874	14120

Table 7: APL2 on SP2. Product of two rectangular matrices (SP2-5)

The asymmetry is better visualized in the following series of plotting, where we used the same scale to facilitate a comparison.

In Figure 1 we plot the data of Table 2, and each line refers to a fixed number of columns.

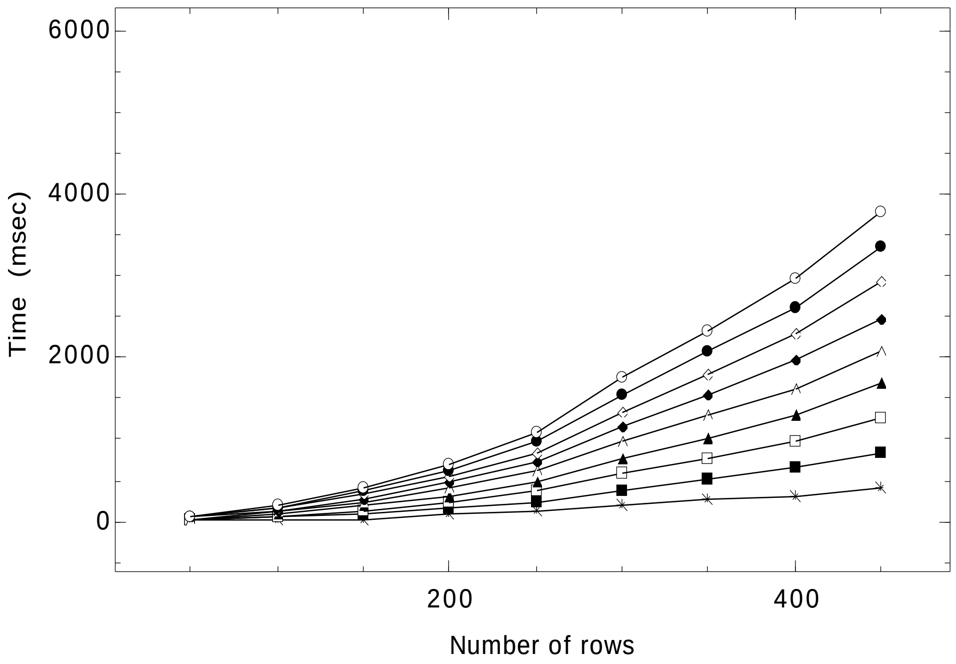

Figure 1. VM-CMS data (each line refers to a fixed number of columns)

In Figure 2 we plot the data of Table 2, but each line now refers to a fixed number of rows.

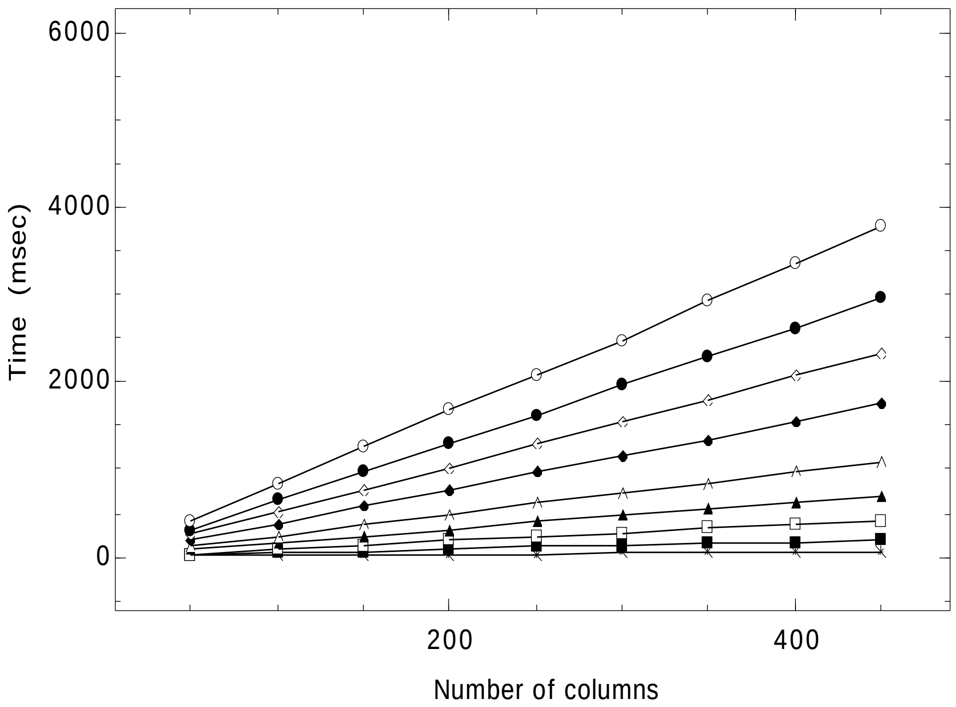

Figure 2. VM-CMS data (each line refers to a fixed number of rows)

In Figure 3 we plot the data of Table 3, and each line refers to a fixed number of columns.

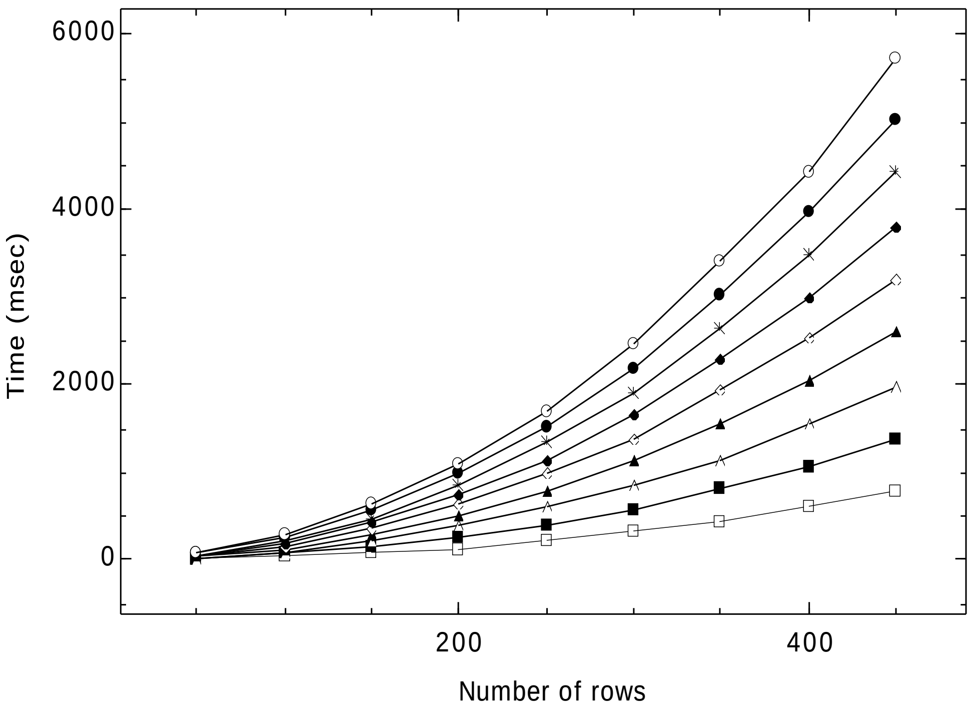

Figure 3. SP1 data (each line refers to a fixed number of columns)

In Figure 4 we plot the data of Table 3, but each line now refers to a fixed number of rows.

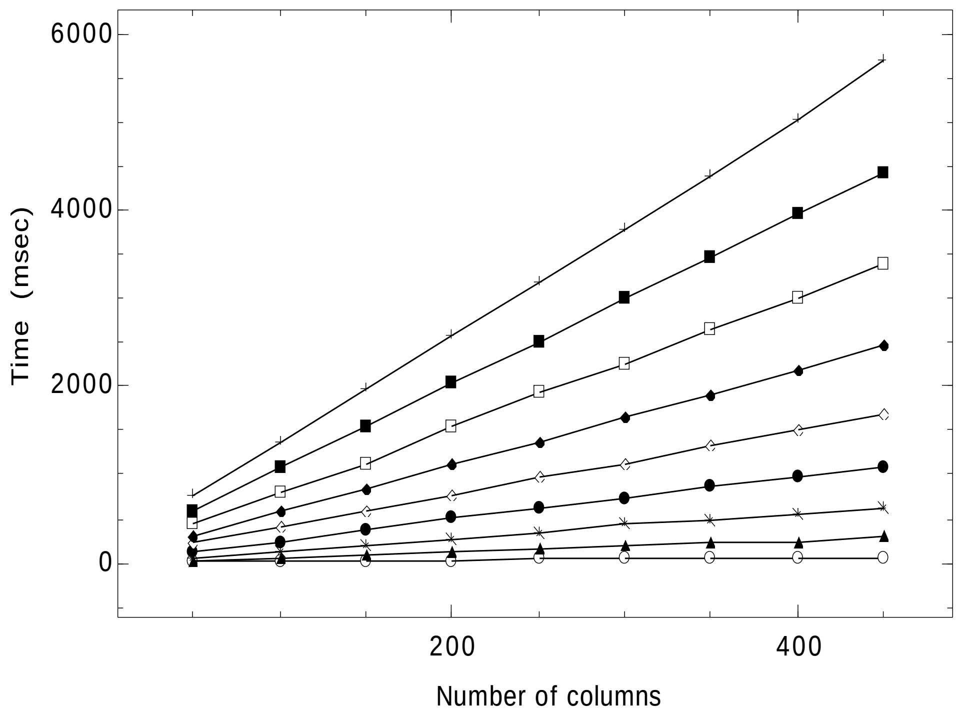

Figure 4. SP1 data (each line refers to a fixed number of rows)

In Figure 5 we plot the data of Table 4, and each line refers to a fixed number of columns.

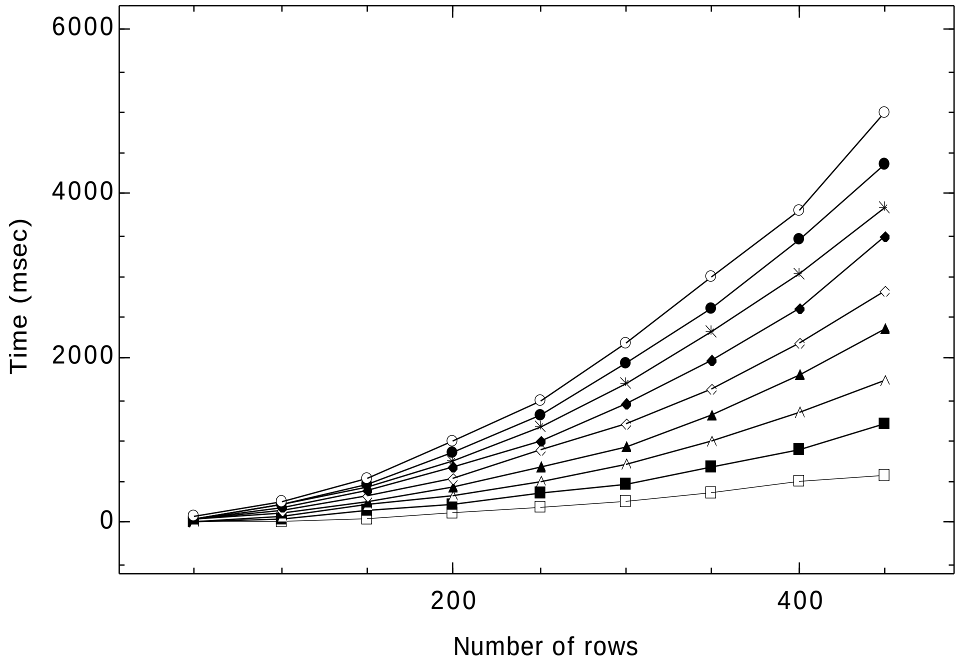

Figure 5. SP2 data (each line refers to a fixed number of columns)

In Figure 6 we plot the data of Table 4, but each line now refers to a fixed number of rows.

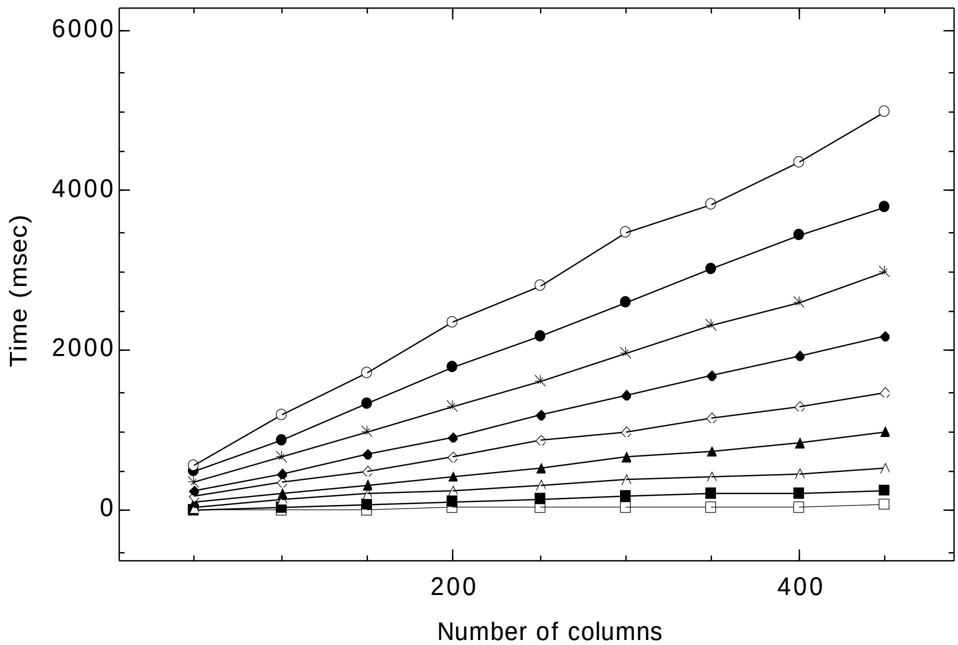

Figure 6. SP2 data (each line refers to a fixed number of rows)

In Figure 7 we plot the data of Table 5, and each line refers to a fixed number of columns.

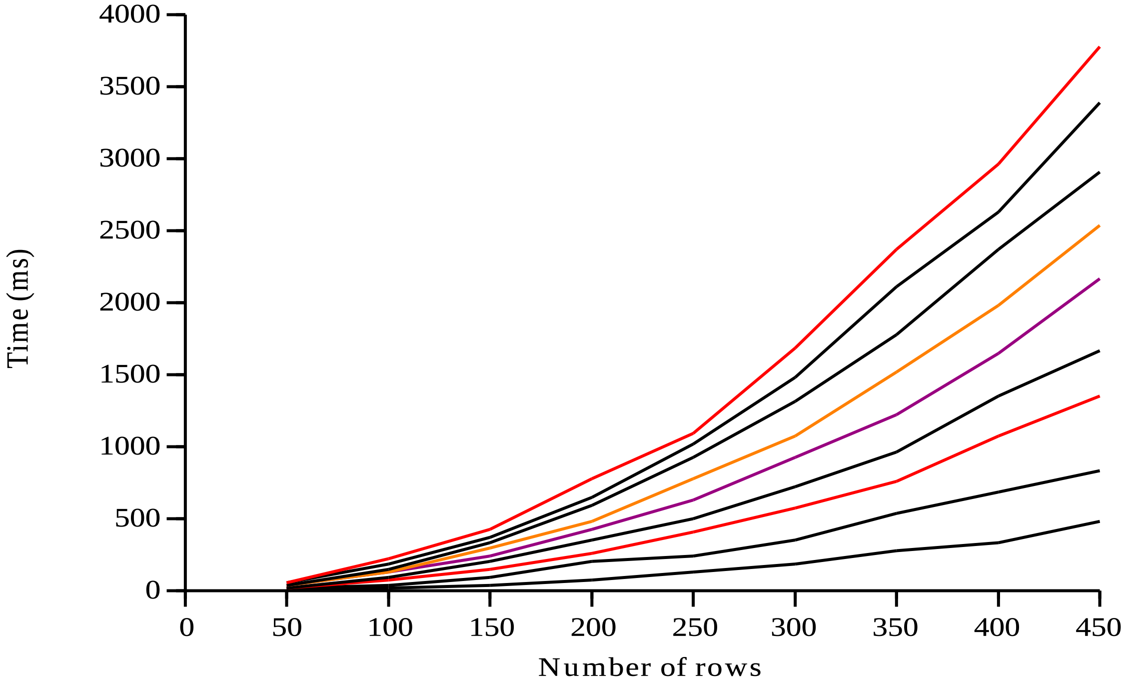

Figure 7. SP2e data (each line refers to a fixed number of columns)

In Figure 8 we plot the data of Table 5, but each line now refers to a fixed number of rows.

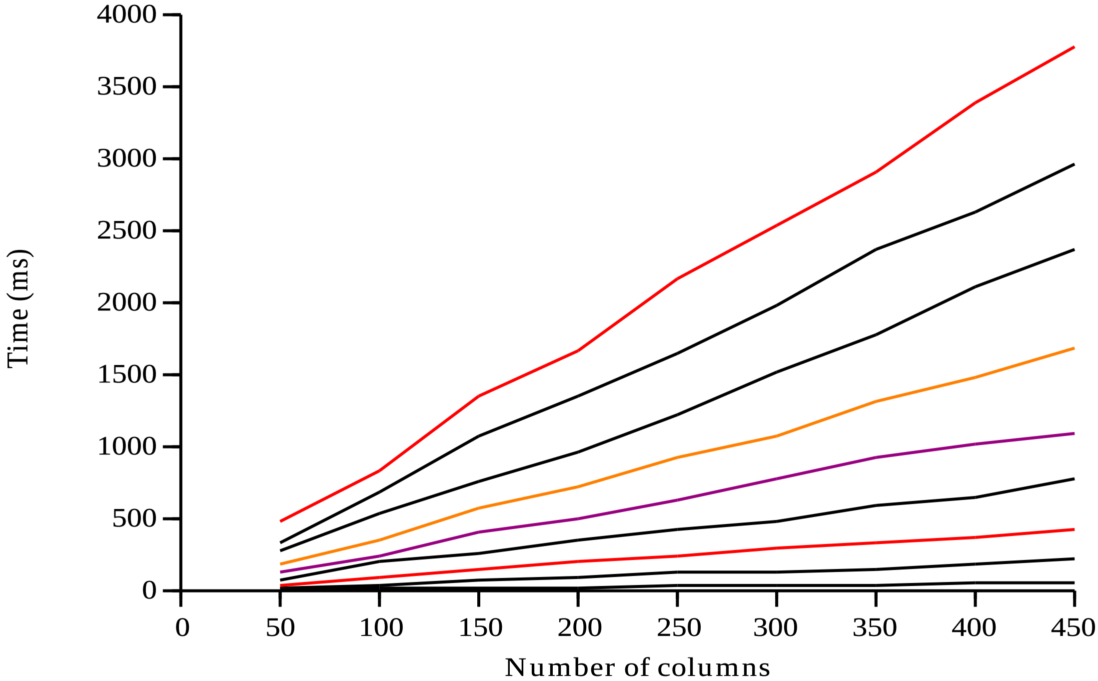

Figure 8. SP2e data (each line refers to a fixed number of rows)

In Figure 9 we plot the data of Table 6, and each line refers to a fixed number of columns.

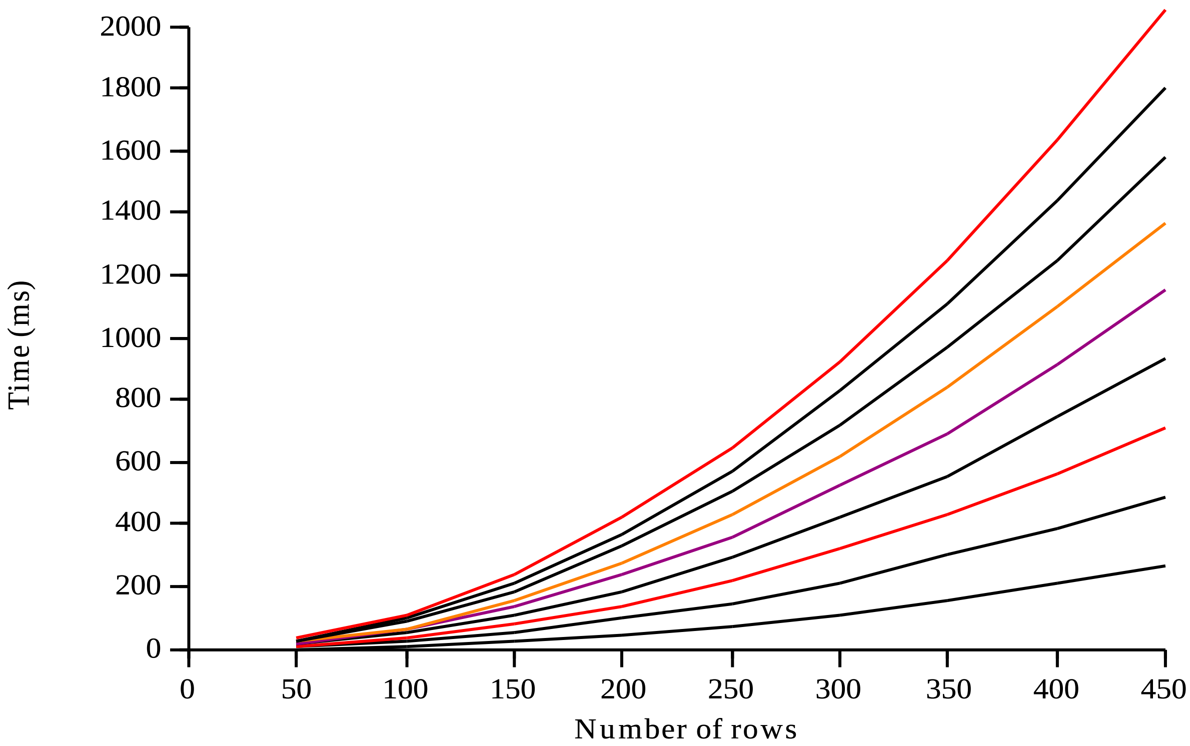

Figure 9. SP2-2 data (each line refers to a fixed number of columns)

In Figure 10 we plot the data of Table 6, but each line now refers to a fixed number of rows.

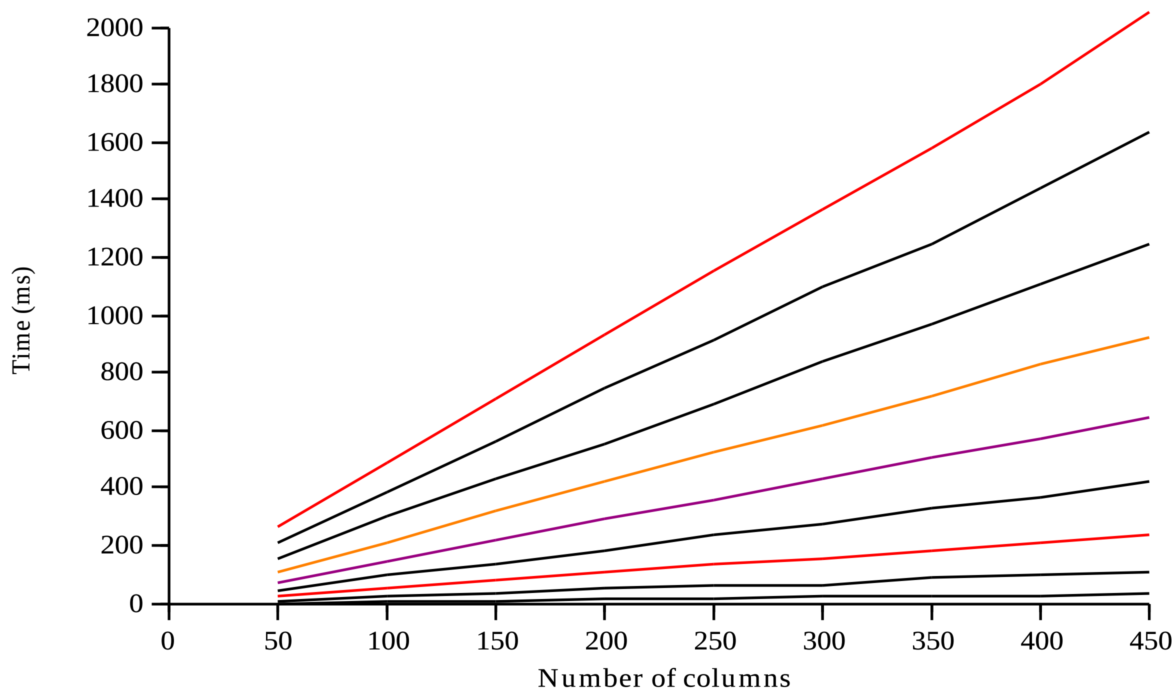

Figure 10. SP2-2 data (each line refers to a fixed number of rows)

In Figure 11 we plot the data of Table 7, and each line refers to a fixed number of columns.

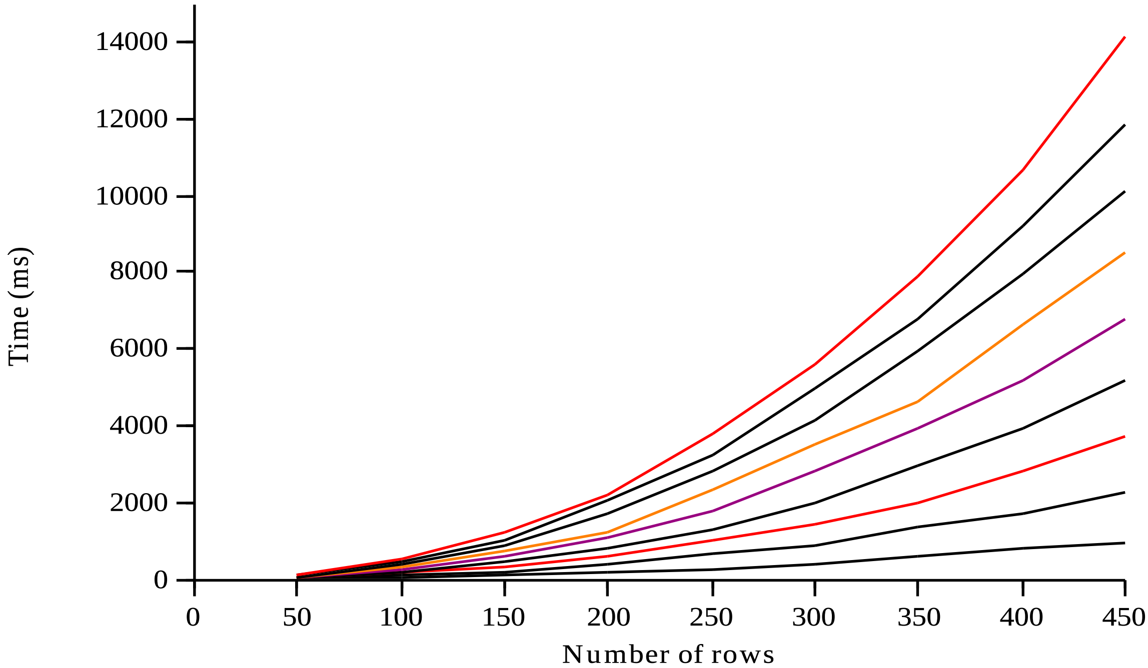

Figure 11. SP2-5 data (each line refers to a fixed number of columns)

In Figure12 we plot the data of Table 7, but each line now refers to a fixed number of rows.

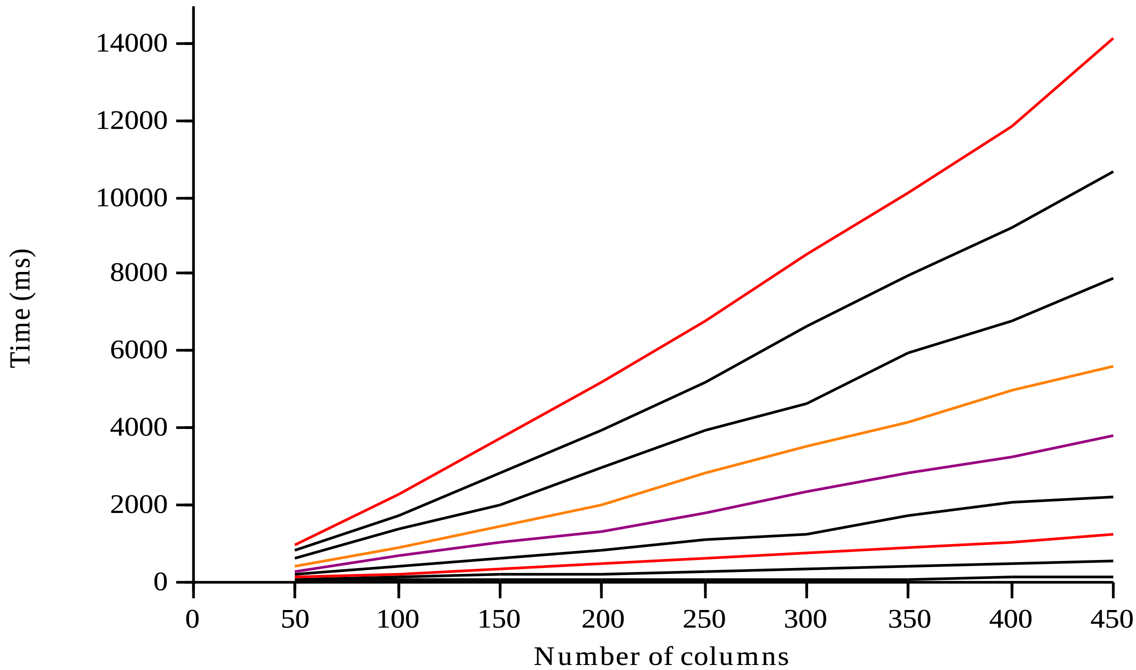

Figure 12. SP2-5 data (each line refers to a fixed number of rows)

System of linear equations

The second set of data (see Tables 8-13) concerns the solution of a system of linear equations; the system has *n* linear equations, and *m* unknowns. APL2 implements this operation as a primitive function. When the matrix of coefficients is rectangular – that is we have more equations than unknowns – the APL2 uses a least squares approximation.

In the Tables *n* labels the rows, and *m* labels the columns. The time of the operation is given in milliseconds.

In Table 8 we have the data obtained on the main-frame IBM, the vector feature was enabled.

	50	100	150	200	250	300	350	400	450
50	49								
100	118	393							
150	199	830	1483						
200	467	1435	2619	3167					
250	827	2263	4005	4485	6951				
300	1082	3177	5939	7421	12077	15531			
350	1344	4202	8545	12494	18422	24703	30388		
400	1543	5176	10959	17481	25737	35838	44370	53830	
450	1801	6523	13890	22949	33998	46267	59373	75458	94106

Table 8: APL2 on VM-CMS, with vector feature. Resolution of a linear system

In Table 9 we have the data obtained on the SP1, and we immediately note a great difference among the square and rectangular matrix cases.

The performance is very good when the number of equations is equal to the number of unknowns; in particular it is better than the vector feature one. With rectangular matrices the performance is very poor, and significantly worse than the vector feature one.

	50	100	150	200	250	300	350	400	450
50	7								
100	910	40							
150	1420	5350	150						
200	1830	7350	16880	360					
250	2440	9290	20970	37350	750				
300	3060	11480	25950	47050	73480	1200			
350	3360	13550	30930	56070	89060	133280	1910		
400	3950	15590	35140	64480	104510	152840	209570	2870	
450	4490	17800	40280	73790	117000	171660	240550	314960	3990

Table 9: APL2 on SP1. Resolution of a linear system

In Table 10 we have the data obtained on the SP2, and we immediately note again the great difference among the square and rectangular matrix cases.

The performance is very good when the number of equations is equal to the number of unknowns; in particular it is better than the the performance of SP1 nodes. With rectangular matrices the performance is still very poor, and significantly worse than the vector feature one. The gain in performance of the SP2 is better for low number of columns, and this fact probably depends on the limited amount of the cash memory.

	50	100	150	200	250	300	350	400	450
50	8								
100	589	34							
150	886	3536	114						
200	1205	4708	10550	260					
250	1526	5888	13250	23413	517				
300	1848	7073	15807	28070	44360	877			
350	2183	8370	18460	33022	51950	74300	1378		
400	2503	9510	21522	38262	63000	85520	116460	2030	
450	2835	10655	24520	42980	67360	95810	131586	181460	2950

Table 10: APL2 on SP2. Resolution of a linear system

In Table 11 we report the data of an enhanced node on the SP2 system.

The performance is very good when the number of equations is equal to the number of unknowns; in particular it is slight better than the previous one. With rectangular matrices the performance is still very poor, and significantly worse than the vector feature one, but it is better than the previous one for great matrices.

	50	100	150	200	250	300	350	400	450
50	3								
100	590	30							
150	880	3540	90						
200	1230	4690	10610	220					
250	1510	5970	12973	23260	450				
300	1740	7196	15550	28170	43590	830			
350	2090	8530	18310	33080	50253	73220	1400		
400	2560	9730	21510	37070	57874	84620	113670	1830	
450	2770	10980	24000	41740	66315	96630	129140	167420	2590

Table 11: APL2 on SP2. Resolution of a linear system (enhanced node)

In Table 12 we report the data of a node of the type labelled as SP2-2.

The performance is very good when the number of equations is equal to the number of unknowns; in particular it is better than the performance of an enhanced node. With rectangular matrices the performance is still significantly worse than the vector feature one, but it is better than the performance of an enhanced node for every matrix dimension.

	50	100	150	200	250	300	350	400	450
50	3								
100	330	17							
150	493	1982	58						
200	650	2912	6024	141					
250	822	3722	7748	14570	268				
300	999	4504	9730	17847	27930	457			
350	1190	5270	11820	21280	33353	48770	720		
400	1480	6077	13757	24636	38598	56590	78221	1069	
450	1673	6860	15581	27990	43833	64030	89271	119280	1500

Table 12: APL2 on SP2-2. Resolution of a linear system

In Table 13 we report the data of a node of the type labelled as SP2-5.

When the number of equations is equal to the number of unknowns the performance is still better than the performance of the main-frame with the vector feature but it is worse than the performance of the SP1 nodes. With rectangular matrices the performance is still significantly worse than the vector feature one, but it is in the order of the performance of the SP2 enhanced nodes.

	50	100	150	200	250	300	350	400	450
50	4								
100	478	45							
150	712	2858	148						
200	981	3820	8682	382					
250	1217	4856	11000	20582	840				
300	1526	6163	13877	25540	40167	1465			
350	1831	7392	16786	30489	49191	72337	2618		
400	2095	8507	19241	34926	56995	83163	114259	4203	
450	2375	9568	21861	40461	64073	93249	130538	173083	6406

Table 13: APL2 on SP2-5. Resolution of a linear system

For these sets of data too we will plot the results in the Figures 13-18.

In Figure 13 we plot the data of Table 8; in Figure 14 we plot the data of Table 9; in Figure 15 we plot the data of Table 10; in Figure 16 we plot the data of Table 11; in Figure 17 we plot the data of Table 12; and in Figure 18 we plot the data of Table 13.

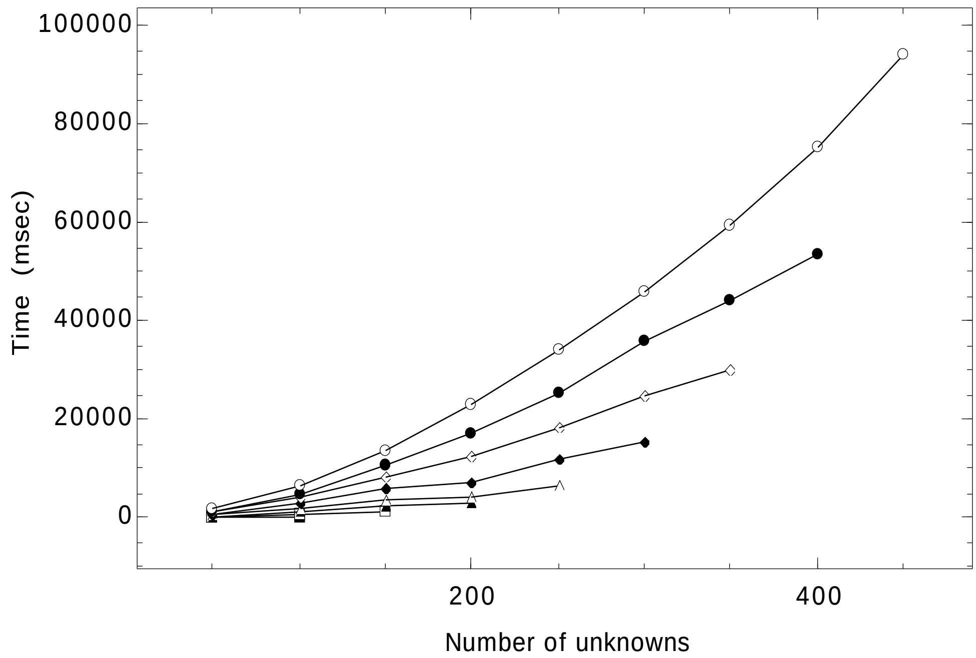

Figure 13. VM data (each line refers to a fixed number of equations)

Roughly speaking the performance shows a relation 1:3:2 among VM with vector feature, SP1, and SP2; as we can see from the ordinate scale.

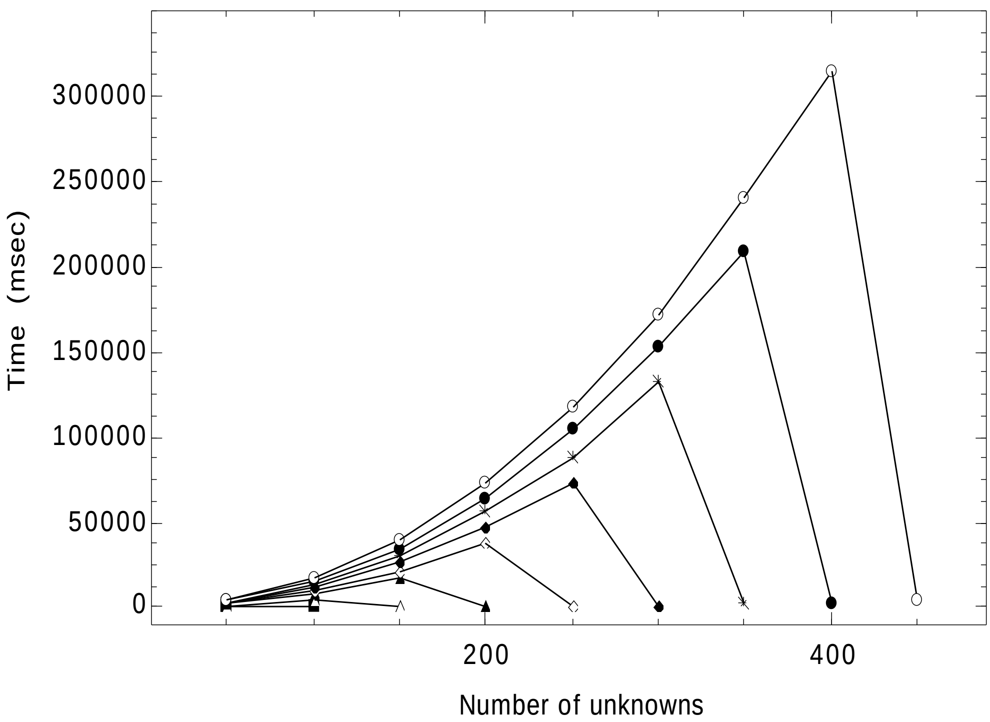

Figure 14. SP1 data (each line refers to a fixed number of equations)

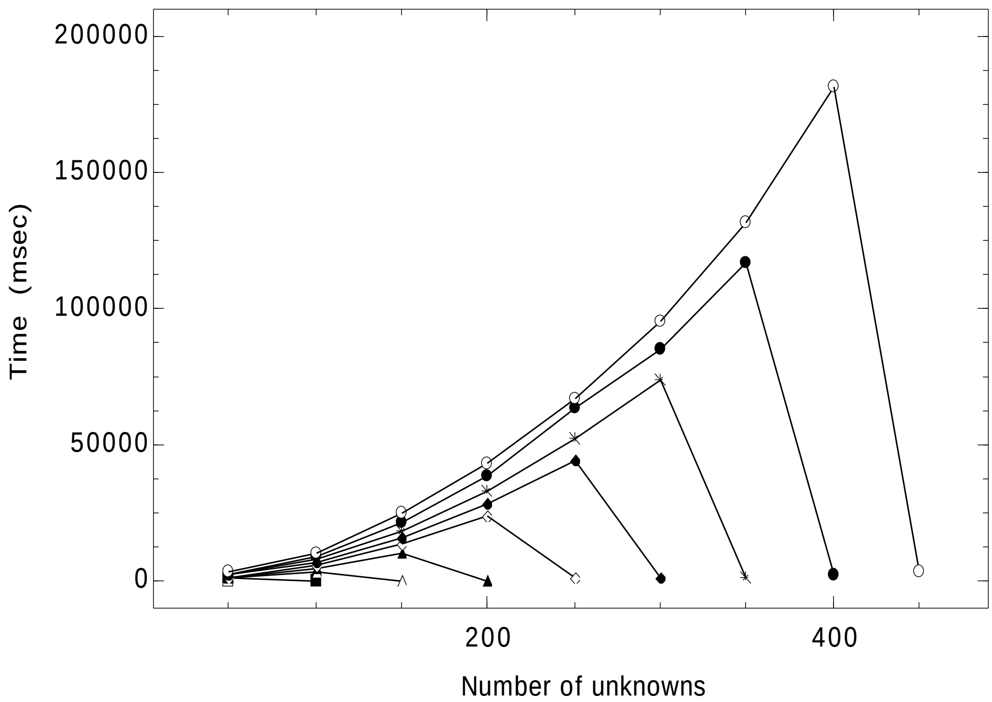

Figure 15. SP2 data (each line refers to a fixed number of equations)

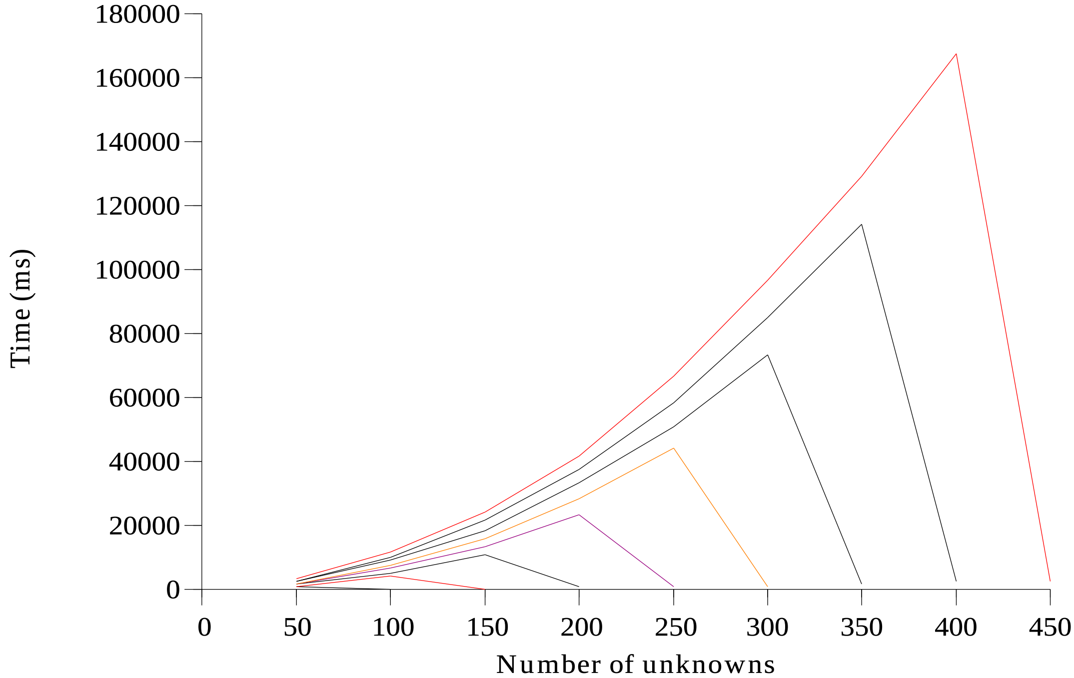

Figure 16. SP2e data (each line refers to a fixed number of equations)

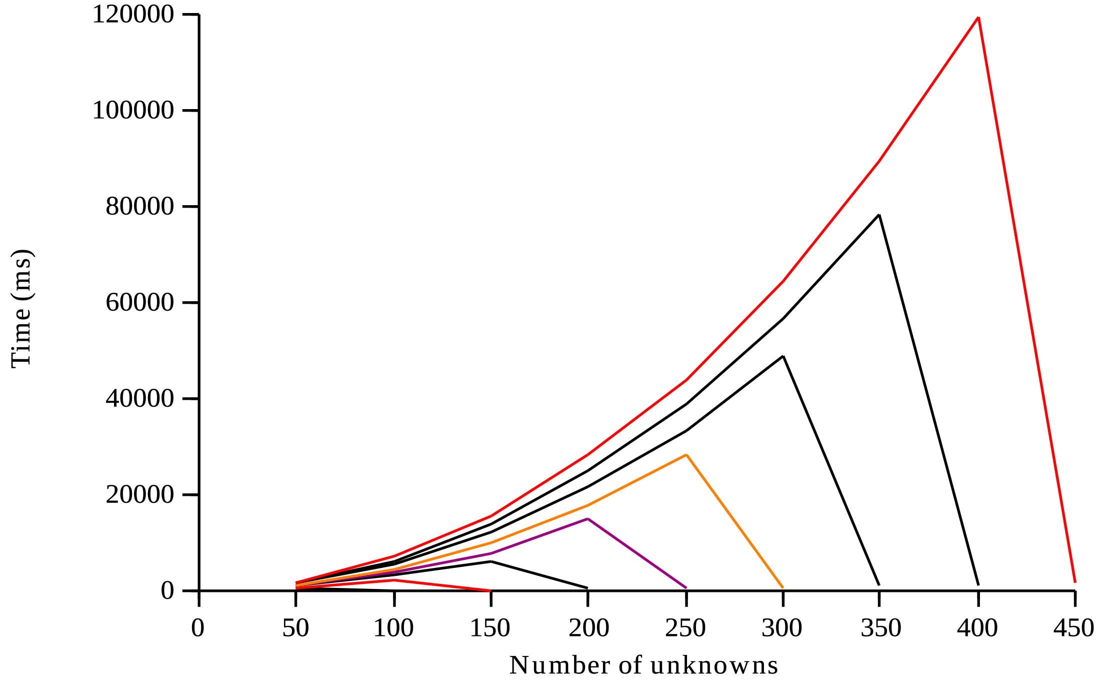

Figure 17. SP2-2 data (each line refers to a fixed number of equations)

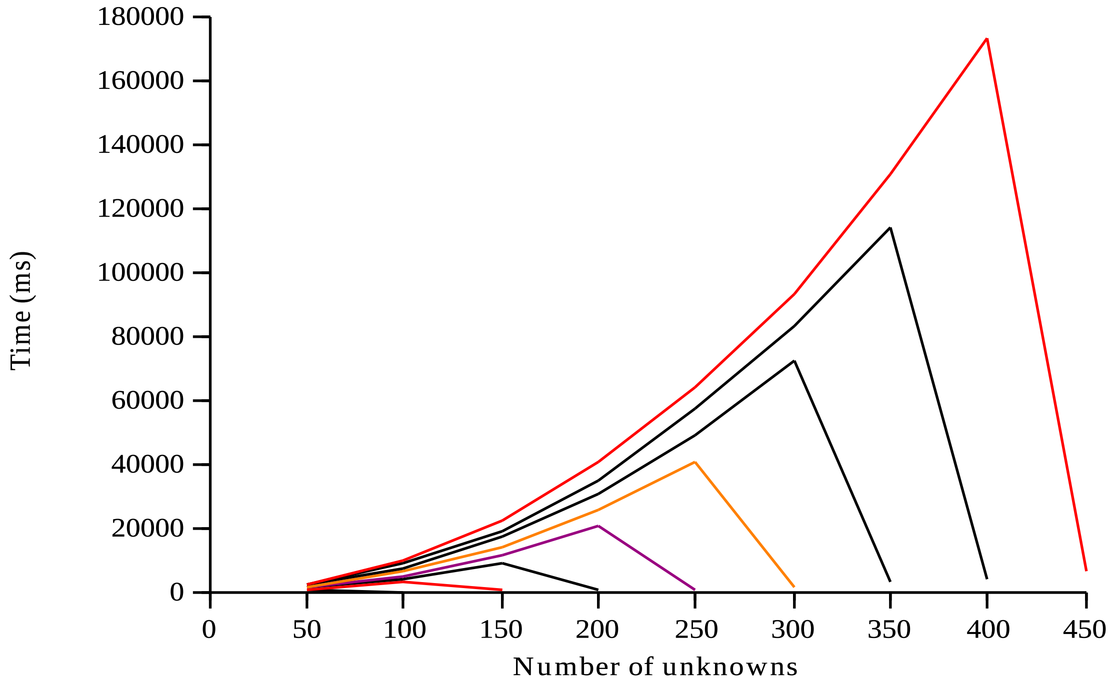

Figure 18. SP2-2 data (each line refers to a fixed number of equations)

Finally in Table 14 we give a comparison between data which concern the use of the supplied function EIGEN to evaluate eigenvalues and eigenvectors of a square real matrix. The first column refers to VM / CMS with the vector feature, and the second one refers to SP2.

The time is in milliseconds and the data refer to square random matrices from 50 to 200 rows and columns.

	50	100	150	200
VM	11770	66967	193050	400199
SP2	27030	187130	649030	1810950
SP2-e	21340	140230	459840	1197170
SP2-2	11730	79580	258340	655395
SP2-5	29150	270273	1150515	3637080

Table 14: APL2 on VM and SP2. Eigenvalues and eigenvectors

The differences are here very evident. The time is higher in SP2 by a factor which spans from 2.3 to 4.52, and this factor increases when the dimension of the matrix increases. The performance is better on the enhanced node, SP2-e row, but in every case it is worse than on the VM with the vector feature. The best performance on the SP2 system is obtained by a node of the type labelled SP2-2; but it is still worse than on the VM with the vector feature, with the exception of small matrices.