

## Appendix A. GraalDOSS implementation

GraalDOSS implementation consists of a public API and an internal implementation. Figure A.1 shows how these components relate in terms of class hierarchy and association. The internal component provides abstract implementations of the public interfaces, in order to make the system modular and extensible. That way, the snapshot heap and snapshot slots can be tuned to support different garbage collector implementations or serve other purposes.

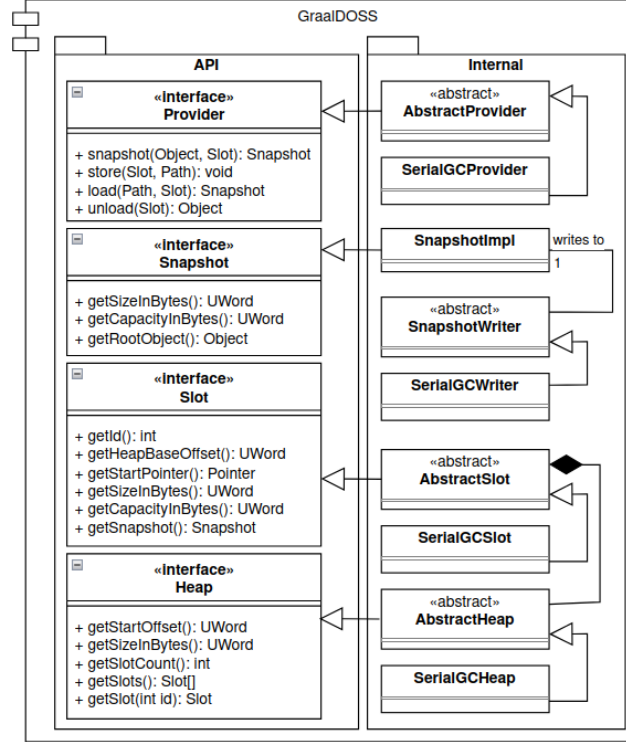


Figure A.1: GraalDoss class UML diagram.

## Appendix B. Detailed microbenchmark results

We group the results into multiple tables based on the operation performed (serialization or deserialization) and the object kinds. Object kinds include primitive types such as integers and doubles (`int`, `double`), strings of length  $n$  (`Str[n]`), primitive arrays of size  $n$  (`int[n]` or `double[n]`), records with  $n$  fields (`R<>[n]`), lists of various types and sizes, hash maps of various types for keys and values with different sizes, and microservice POJOs, e.g., `Client` (abbr. `Client`). The breakdown of the `Client` POJO fields is shown in Table B.1.

Tables B.2 shows the serialization throughput for arrays of primitive types, while B.3 shows the serialization throughput for strings, records and POJOs. Tables B.4, B.5, and B.6 show serialization throughput for our list and map workloads. Throughput is measured in the number of serialization operations per second (ops/s, higher is better). Sizes correspond to serialized object sizes in JSON, binary (`Bin`), or a snapshot (`Snap`) generated by GraalDOSS, measured in bytes. Similarly, tables B.7 and B.8 show deserialization throughput for arrays of primitive types, strings, records and POJOs, while tables B.9, B.10, and B.11 show deserialization throughput for list and map workloads.

S/D framework versions used in the experiments can be found in Table B.12.

**Table B.1**

Breakdown of the Client POJO fields.

Field (type)	Description
id (long)	Client identifier.
index (int)	Work contract index.
isActive (boolean)	Account activity.
balance (BigDecimal)	Financial balance.
picture (String)	Picture URL (max 100 characters).
age (int)	Age.
eyeColor (EyeColor)	Eye color, enum, one of: BROWN, BLUE, or GREEN.
name (String)	Full name (max 20 characters).
gender (String)	Gender (max 20 characters).
company (String)	Company name (max 20 characters).
emails (String[])	Email list (max 10 items, max 20 characters each).
phones (long[])	Phone number list (max 10 items).
address (String)	Address (max 20 characters).
about (String)	Additional information (max 20 characters).
registered (LocalDate)	Registration date.
latitude (double)	Latitude.
longitude (double)	Longitude.
tags (List<String>)	Custom tags (max 50 items, max 10 characters each).
partners (List<Partner>)	List of partners (max 30 items). Partner is a record consisting of: ID (long), name (String), and partnership date (OffsetDateTime).

**Table B.2**Serialization throughput for arrays of primitive types with length  $n$  (int[n] and double[n]).

Object kind	Size (B)			java [2]	jackson [3]	gson [4]	dsljson [5]	fastjson2 [6]	Kryo [1]		Gaal DOSS
	JSON	Bin	Snap						def	ref	
int[ 64]	701	381	824	72 876	383 293	106 259	355 606	458 475	<b>482 214</b>	366 439	88 325
int[ 128]	1417	768	1592	38 626	209 828	52 869	176 585	237 707	<b>246 108</b>	193 174	60 317
int[ 256]	2809	1522	3128	18 863	102 367	27 056	88 381	119 573	<b>126 163</b>	96 301	34 875
int[ 512]	5628	3036	6200	10 219	55 966	13 373	47 622	59 488	<b>62 862</b>	48 777	20 075
int[1024]	11 271	6078	12 344	4915	25 616	6833	23 526	29 589	<b>31 635</b>	24 755	10 941
int[2048]	22 568	12 174	24 632	2422	11 067	3082	10 965	13 031	<b>15 760</b>	12 451	5650
int[4096]	44 959	24 333	49 208	1262	5306	1650	4913	5668	<b>7894</b>	6208	2877
double[ 64]	1232	578	1336	74 589	122 463	74 097	128 149	68 435	<b>526 384</b>	387 222	90 194
double[ 128]	2469	1155	2616	37 881	61 992	37 992	62 343	34 625	<b>266 194</b>	200 838	56 318
double[ 256]	4920	2307	5176	18 577	30 961	18 905	32 438	17 314	<b>135 400</b>	102 350	37 574
double[ 512]	9863	4611	10 296	9227	15 590	9163	16 077	8545	<b>67 783</b>	52 383	21 012
double[1024]	19 746	9219	20 536	4889	7394	4602	8178	4265	<b>34 003</b>	25 671	11 361
double[2048]	39 463	18 435	41 016	2263	3463	2224	3885	2100	<b>17 172</b>	13 079	5823
double[4096]	78 898	36 867	81 976	1271	1277	1057	2011	1037	<b>8570</b>	6526	3101

**Table B.3**

Serialization throughput for strings of length  $n$  ( $\text{Str}[n]$ ), records with  $n$  primitive and string fields ( $\text{R}<\text{int}>[n]$ ,  $\text{R}<\text{double}>[n]$ ,  $\text{R}<\text{Str}[32]>[n]$ ), and the Client POJO ( $\text{Client}$ ).

Object kind	Size (B)			java [2]	jackson [3]	gson [4]	dsljson [5]	fastjson2 [6]	Kryo [1]		Gaal DOSS
	JSON	Bin	Snap						def	ref	
$\text{Str}[32]$	34	33	104	2 241 883	2 754 863	2 369 467	3 328 652	3 915 962	<b>5 288 242</b>	3 938 018	163 558
$\text{Str}[64]$	66	67	136	1 709 166	2 282 946	1 677 156	2 242 981	2 397 117	<b>3 268 123</b>	2 746 566	164 441
$\text{Str}[128]$	130	131	200	1 217 154	1 728 841	1 126 503	1 534 674	1 323 160	<b>1 805 080</b>	1 610 783	150 201
$\text{Str}[256]$	258	259	328	768 092	933 839	630 088	852 175	710 302	<b>987 798</b>	914 266	158 663
$\text{Str}[512]$	514	515	584	532 695	<b>635 679</b>	353 321	468 058	377 507	511 301	488 447	160 607
$\text{R}<\text{int}>[2]$	34	11	64	90 422	1 841 817	1 176 691	3 031 141	3 927 055	<b>5 695 287</b>	3 577 847	167 974
$\text{R}<\text{int}>[4]$	64	21	72	88 565	1 462 874	668 767	2 024 897	2 567 878	<b>3 817 212</b>	2 618 882	171 947
$\text{R}<\text{int}>[8]$	128	39	88	82 724	956 442	364 036	1 264 176	1 528 011	<b>2 318 674</b>	1 699 181	170 047
$\text{R}<\text{int}>[16]$	265	79	120	80 345	649 660	180 676	745 516	829 724	<b>1 304 425</b>	987 960	169 461
$\text{R}<\text{double}>[2]$	50	17	72	91 233	1 115 429	955 713	1 989 719	1 540 758	<b>6 184 348</b>	3 664 740	168 498
$\text{R}<\text{double}>[4]$	98	33	88	87 541	781 492	541 215	1 231 020	842 728	<b>4 126 507</b>	2 690 437	170 842
$\text{R}<\text{double}>[8]$	195	65	120	82 873	474 059	300 174	696 656	442 229	<b>2 271 038</b>	1 714 764	168 362
$\text{R}<\text{double}>[16]$	399	129	184	81 415	272 412	152 431	357 116	219 439	<b>1 309 864</b>	966 526	153 952
$\text{R}<\text{Str}[32]>[2]$	81	67	176	82 312	1 321 182	855 561	1 784 800	1 849 155	<b>2 331 535</b>	1 812 089	145 895
$\text{R}<\text{Str}[32]>[4]$	161	133	296	80 141	883 566	479 465	1 044 143	1 022 215	<b>1 270 820</b>	1 039 862	143 886
$\text{R}<\text{Str}[32]>[8]$	321	265	536	71 961	549 762	264 385	580 113	538 678	<b>655 664</b>	555 273	139 099
$\text{R}<\text{Str}[32]>[16]$	648	529	1016	60 961	297 934	136 412	306 498	276 474	<b>350 095</b>	291 343	117 780
Client	4428	971	3536	22 855	61 848	23 198	125 557	65 900	<b>224 808</b>	59 968	48 293

Table B.4

Serialization throughput for lists of variable types and sizes.

Type	Size	Size (B)			java [2]	jackson [3]	gson [4]	dsljson [5]	fastjson2 [6]	Kryo [1]		Gaal DOSS
		JSON	Bin	Snap						def	ref	
int	16	172	84	272	211 222	973 584	390 972	1 230 444	1 391 217	<b>1 709 295</b>	1 216 789	139 718
	32	351	163	464	129 301	572 663	199 174	641 337	767 759	<b>925 683</b>	679 103	109 308
	64	707	323	848	69 401	325 918	109 853	368 162	407 857	<b>475 298</b>	372 589	86 624
	128	1400	637	1616	36 342	181 021	52 355	176 781	209 401	<b>253 383</b>	197 637	58 178
	256	2822	1272	3152	18 061	100 511	24 355	92 895	103 874	<b>133 223</b>	100 929	33 362
	512	5661	2533	6224	9484	47 151	12 319	46 835	52 655	<b>66 349</b>	51 661	19 392
double	16	310	132	400	208 689	388 654	257 188	447 714	266 826	<b>1 751 841</b>	1 252 367	137 192
	32	622	260	720	116 398	194 145	152 567	229 116	134 999	<b>1 010 276</b>	726 890	114 500
	64	1242	517	1360	66 989	115 522	64 136	117 541	68 377	<b>534 489</b>	399 355	88 985
	128	2464	1029	2640	37 993	60 743	38 458	59 376	34 051	<b>285 069</b>	211 767	57 251
	256	4954	2053	5200	18 212	30 473	16 981	31 035	16 768	<b>143 115</b>	108 551	37 074
	512	9839	4101	10 320	8967	15 653	8275	15 473	8243	<b>73 406</b>	55 051	20 912
Str[32]	16	561	532	1040	194 866	<b>439 221</b>	193 314	375 553	322 901	401 206	330 588	117 012
	32	1121	1060	2000	98 910	<b>238 716</b>	107 159	205 065	168 737	206 942	173 598	85 091
	64	2241	2117	3920	54 646	<b>125 536</b>	49 590	104 890	85 276	104 659	84 687	56 145
	128	4481	4229	7760	26 092	<b>62 890</b>	26 001	50 770	43 190	52 722	43 210	34 078
	256	8961	8453	15 440	14 127	<b>30 678</b>	13 359	25 971	21 581	26 650	21 749	19 145
	512	17 921	16 901	30 800	6871	<b>15 440</b>	6717	12 936	10 848	13 349	10 723	10 552
R<int>[2]	16	537	180	400	5172	257 034	79 543	237 037	357 612	<b>463 070</b>	309 381	125 402
	32	1107	351	720	2618	146 666	38 019	117 457	179 090	<b>238 337</b>	159 150	108 496
	64	2175	703	1360	1297	74 913	19 800	65 377	89 676	<b>121 913</b>	82 301	85 257
	128	4368	1400	2640	650	37 547	9834	33 906	44 672	<b>62 062</b>	42 588	53 780
	256	8690	2783	5200	321	19 250	4592	14 794	21 642	31 254	20 718	<b>35 001</b>
	512	17 373	5578	10 320	165	8993	2351	7536	10 694	15 711	10 680	<b>19 853</b>
R<int>[16]	16	4245	1290	1296	4998	49 936	11 824	44 336	54 110	80 653	65 507	<b>128 916</b>
	32	8462	2565	2512	2599	22 974	5901	24 182	27 048	40 663	33 289	<b>95 323</b>
	64	16 961	5139	4944	1273	11 308	2913	11 045	12 983	20 406	16 564	<b>75 558</b>
	128	33 915	10 261	9808	645	5449	1473	5320	6026	10 226	8352	<b>47 228</b>
	256	67 792	20 476	19 536	326	1882	728	2681	2934	5104	4248	<b>27 190</b>
	512	135 413	40 955	38 992	159	835	346	1189	1425	2574	2070	<b>15 535</b>
R<double>[2]	16	807	276	528	5095	125 336	62 650	142 450	111 239	463 835	312 511	<b>131 868</b>
	32	1611	548	976	2568	65 827	31 712	75 303	55 091	236 733	166 091	<b>116 246</b>
	64	3245	1093	1872	1307	35 117	17 293	36 465	27 254	124 964	83 795	<b>90 269</b>
	128	6479	2181	3664	655	17 957	8051	17 361	13 611	61 023	43 527	<b>52 895</b>
	256	12 944	4357	7248	320	8661	4008	8631	6935	31 345	21 584	<b>31 934</b>
	512	25 833	8709	14 416	165	4504	1978	4259	3407	15 922	10 700	<b>18 732</b>
R<double>[16]	16	6356	2068	2320	5104	19 232	9871	22 152	13 935	82 772	69 263	<b>109 734</b>
	32	12 745	4132	4560	2480	9600	4907	11 563	6886	43 576	33 762	<b>91 517</b>
	64	25 418	8261	9040	1282	4650	2173	5260	3457	21 772	17 261	<b>63 551</b>
	128	50 832	16 517	18 000	640	2287	1240	2815	1676	11 076	8531	<b>39 409</b>
	256	101 646	33 029	35 920	319	887	484	1412	849	5520	4290	<b>22 979</b>
	512	203 588	66 053	71 760	156	386	286	640	415	2797	2178	<b>12 637</b>
R<Str[32]>[2]	16	1313	1076	2192	4973	153 136	61 283	125 457	139 818	<b>166 292</b>	118 716	67 227
	32	2625	2148	4304	2446	75 328	30 705	65 057	71 044	<b>84 104</b>	61 596	48 273
	64	5249	4293	8528	1221	39 412	15 201	32 726	34 661	<b>42 560</b>	29 855	28 370
	128	10 497	8581	16 976	612	20 023	7747	16 380	17 870	<b>21 438</b>	16 020	16 354
	256	20 993	17 157	33 872	306	10 086	4025	7961	8802	<b>10 790</b>	7936	9181
	512	41 985	34 309	67 664	151	5016	1834	4280	4413	<b>5383</b>	3790	4546
R<Str[32]>[16]	16	10 385	8468	15 632	3592	21 710	8924	19 843	17 686	<b>22 030</b>	17 683	18 475
	32	20 769	16 932	31 184	1853	<b>11 132</b>	4615	9267	8864	10 994	9027	9777
	64	41 537	33 861	62 288	921	5308	2288	4660	4390	<b>5461</b>	4440	5111
	128	83 073	67 717	124 496	465	1708	1164	2463	2191	<b>2772</b>	2245	2654
	256	166 145	135 429	248 912	235	767	512	1112	1104	<b>1371</b>	1022	1355
	512	332 289	270 853	497 744	115	380	254	567	546	<b>690</b>	477	660
Client	16	34 989	22 196	69 136	2157	3763	1398	4869	4657	<b>7950</b>	4392	3605
	32	76 469	47 551	147 256	1062	1193	766	1890	2471	<b>2887</b>	2270	2008
	64	164 966	93 200	263 656	548	521	349	913	1322	<b>1693</b>	1081	940
	128	321 029	177 247	520 200	285	254	184	426	608	<b>776</b>	437	462
	256	645 133	375 020	1 059 440	133	127	91	216	309	<b>384</b>	203	143
	512	1 317 583	765 983	2 059 624	62	61	44	55	156	<b>198</b>	101	53

Table B.5

Serialization throughput for maps that have integer keys.

Map (Key type: int)		Size (B)			java	jackson	gson	dsljson	fastjson2	Kryo [1]		Graal
Value type	Size	JSON	Bin	Snap	[2]	[3]	[4]	[5]	[6]	def	ref	DOSS
int	16	378	192	864	125 739	329 551	150 532	602 847	634 330	<b>807 345</b>	624 029	86 006
	32	773	378	1632	69 778	180 311	72 962	327 795	334 056	<b>402 604</b>	300 519	60 775
	64	1534	757	3168	36 859	90 366	39 231	182 356	174 076	<b>195 661</b>	160 474	35 798
	128	3080	1522	6240	19 135	49 071	18 550	87 185	88 670	<b>96 382</b>	78 559	20 861
	256	6158	3050	12 384	8943	25 122	9187	47 288	44 307	<b>47 359</b>	38 564	11 202
	512	12 303	6082	24 672	4444	11 742	4563	21 981	20 881	<b>23 096</b>	19 260	6215
Str[8]	16	389	257	1248	124 807	301 891	145 507	486 001	<b>515 005</b>	485 191	356 563	76 547
	32	770	511	2400	68 232	151 620	76 117	258 121	<b>270 656</b>	235 151	183 812	46 551
	64	1528	1024	4704	33 573	75 775	39 187	125 688	<b>137 820</b>	117 881	90 608	29 170
	128	3067	2046	9312	16 954	38 724	18 250	66 813	<b>68 780</b>	59 434	44 011	17 640
	256	6127	4081	18 528	8977	20 177	9593	31 769	<b>33 702</b>	29 789	21 431	9323
	512	12 300	8154	36 960	4302	9449	4693	<b>16 191</b>	15 894	14 636	10 891	4910
Str[32]	16	769	640	1632	99 786	210 413	117 191	275 986	251 340	<b>278 966</b>	230 822	78 871
	32	1542	1280	3168	53 679	119 467	55 416	138 304	126 988	<b>142 102</b>	114 414	46 561
	64	3077	2558	6240	28 621	58 446	29 544	70 120	63 334	<b>70 942</b>	57 073	27 380
	128	6147	5116	12 384	13 616	28 952	15 382	<b>37 154</b>	31 138	35 383	28 391	16 920
	256	12 297	10 218	24 672	6687	14 198	7616	<b>18 335</b>	15 333	17 620	14 138	9402
	512	24 563	20 443	49 248	3523	7599	3679	<b>9088</b>	7567	8941	7290	4724
R<int>[2]	16	750	286	992	4961	159 476	58 612	224 436	283 974	<b>308 280</b>	222 224	82 193
	32	1503	574	1888	2492	79 818	29 088	116 876	144 182	<b>151 303</b>	108 341	60 716
	64	2995	1144	3680	1260	40 898	15 001	62 644	73 831	<b>77 727</b>	54 474	34 350
	128	6016	2286	7264	624	20 310	7120	29 916	36 210	<b>37 358</b>	26 970	21 250
	256	11 990	4565	14 432	310	10 315	3692	15 583	17 221	<b>19 066</b>	13 743	10 992
	512	24 001	9103	28 768	154	4921	1854	7372	7950	<b>9510</b>	6964	6153
R<int>[16]	16	4450	1397	1888	4780	43 645	11 143	46 854	52 750	71 176	61 960	<b>84 232</b>
	32	8875	2789	3680	2471	20 573	5494	21 837	25 726	36 670	29 662	<b>54 517</b>
	64	17 759	5550	7264	1241	10 223	2773	10 917	12 369	18 251	15 138	<b>32 125</b>
	128	35 546	11 123	14 432	601	5074	1413	5164	5660	9291	7507	<b>18 371</b>
	256	71 005	22 240	28 768	305	1791	685	2622	2772	4647	3736	<b>10 425</b>
	512	142 212	44 500	57 440	152	746	323	1158	1330	2308	1834	<b>5355</b>
R<double>[2]	16	1021	386	1120	4841	98 145	50 856	124 256	96 200	<b>291 950</b>	212 208	80 887
	32	2025	767	2144	2480	49 834	25 580	67 919	48 713	<b>149 467</b>	106 310	55 002
	64	4065	1527	4192	1247	25 671	12 968	34 063	24 321	<b>74 411</b>	53 872	36 086
	128	8110	3063	8288	628	12 562	6339	16 113	12 046	<b>37 175</b>	26 679	20 776
	256	16 266	6134	16 480	312	6106	3220	7944	6015	<b>18 213</b>	13 271	11 367
	512	32 488	12 249	32 864	156	3290	1642	4211	2944	<b>9149</b>	6575	5957
R<double>[16]	16	6571	2175	2912	4821	18 887	8907	22 931	13 793	<b>74 807</b>	59 867	74 545
	32	13 118	4353	5728	2374	9076	4178	11 309	6717	37 189	30 171	<b>53 642</b>
	64	26 262	8700	11 360	1222	4573	2062	5368	3351	19 101	14 926	<b>31 098</b>
	128	52 552	17 400	22 624	616	2071	1194	2639	1677	9558	7323	<b>17 207</b>
	256	105 046	34 803	45 152	298	864	469	1314	815	4884	3678	<b>9313</b>
	512	210 109	69 606	90 208	152	369	276	623	403	2411	1850	<b>4776</b>
R<Str[32]>[2]	16	1521	1186	2784	4721	104 851	50 571	119 027	117 316	<b>136 765</b>	103 075	50 424
	32	3041	2368	5472	2378	57 153	23 689	58 511	58 351	<b>68 397</b>	51 396	33 556
	64	6085	4737	10 848	1189	28 947	12 814	29 253	29 465	<b>34 319</b>	25 969	18 463
	128	12 163	9471	21 600	589	14 755	6138	15 470	14 551	<b>16 955</b>	12 525	10 423
	256	24 297	18 934	43 104	289	6926	3052	7729	7321	<b>8395</b>	6330	5354
	512	48 625	37 845	86 112	146	3354	1565	3633	3680	<b>4289</b>	3170	2775
R<Str[32]>[16]	16	10 593	8577	16 224	3556	19 846	8119	19 507	17 165	<b>21 454</b>	17 440	16 352
	32	21 188	17 154	32 352	1725	9721	4443	9799	8686	<b>10 778</b>	8621	8640
	64	42 375	34 302	64 608	886	5005	2048	4610	4334	<b>5293</b>	4332	4758
	128	84 728	68 601	129 120	442	1462	904	2440	2147	<b>2664</b>	2159	2402
	256	169 475	137 208	258 144	218	730	461	1088	1081	<b>1335</b>	967	1240
	512	338 938	274 412	516 192	113	362	224	526	529	<b>667</b>	470	594
Client	16	42 306	25 173	54 688	2131	3839	1657	4246	4352	<b>5904</b>	5365	3686
	32	95 290	44 475	138 792	1091	946	807	2142	2549	<b>2906</b>	2528	1502
	64	158 375	93 542	266 256	515	487	386	926	1189	<b>1621</b>	995	878
	128	332 475	189 197	541 360	268	269	184	462	633	<b>751</b>	461	422
	256	679 066	368 784	1 051 568	133	122	89	216	320	<b>416</b>	217	141
	512	1 291 259	750 788	2 145 200	65	61	45	56	150	<b>201</b>	102	55

Table B.6

Serialization throughput for maps that have strings of length 8 as keys.

Map (Key type: Str[8])		JSON	Size (B) Bin	Snap	java [2]	jackson [3]	gson [4]	dsljson [5]	fastjson2 [6]	Kryo [1]		Gaal DOSS
Value type	Size									def	ref	
int	16	349	257	1248	123 577	454 082	171 609	508 278	<b>532 540</b>	468 579	360 978	75 807
	32	702	512	2400	69 267	237 925	92 859	256 536	<b>286 623</b>	234 617	182 293	49 563
	64	1412	1024	4704	34 049	127 766	47 034	132 404	<b>144 626</b>	118 729	90 723	29 123
	128	2805	2042	9312	17 195	64 408	22 616	70 852	<b>73 111</b>	58 259	43 468	17 504
	256	5623	4081	18 528	8798	33 948	11 731	35 348	<b>35 435</b>	29 235	21 741	9421
	512	11 238	8150	36 960	4355	16 412	5462	16 301	<b>16 634</b>	14 566	10 843	5029
Str[8]	16	353	322	1632	143 624	357 731	163 565	381 517	<b>447 498</b>	392 678	283 864	72 310
	32	705	642	3168	78 081	192 511	81 270	207 108	<b>229 504</b>	197 097	142 799	44 143
	64	1409	1282	6240	40 704	106 020	43 011	107 347	<b>111 749</b>	100 907	70 763	26 346
	128	2817	2563	12 384	20 469	51 810	21 989	54 460	<b>56 141</b>	49 344	33 789	15 517
	256	5633	5123	24 672	9998	26 126	10 467	26 243	<b>27 669</b>	24 211	16 413	8156
	512	11 265	10 243	49 248	4975	13 881	5721	13 309	<b>13 554</b>	12 168	8522	4292
Str[32]	16	737	706	2016	121 667	252 365	119 407	<b>266 866</b>	235 505	247 589	195 049	73 165
	32	1473	1410	3936	64 130	<b>144 002</b>	63 089	134 697	118 770	126 767	98 639	42 811
	64	2945	2818	7776	31 120	<b>70 063</b>	32 853	65 882	59 210	63 896	49 129	25 752
	128	5889	5635	15 456	14 819	<b>35 304</b>	16 337	34 502	29 329	32 072	24 174	15 095
	256	11 777	11 267	30 816	7513	<b>18 134</b>	8629	17 386	14 790	15 966	12 300	7876
	512	23 553	22 531	61 536	3531	<b>9098</b>	3969	8694	7348	7893	6021	4159
R<int>[2]	16	718	351	1376	4992	171 218	60 918	162 321	<b>251 007</b>	248 462	170 422	82 526
	32	1450	703	2656	2481	92 181	31 867	80 678	<b>130 451</b>	125 307	87 194	48 974
	64	2881	1397	5216	1248	48 955	15 189	38 271	<b>64 073</b>	61 929	44 323	30 824
	128	5744	2806	10 336	622	24 760	7924	19 638	29 880	<b>31 656</b>	21 880	17 381
	256	11 507	5596	20 576	312	12 250	4025	9695	14 708	<b>15 823</b>	11 040	9378
	512	23 064	11 195	41 056	152	6178	2011	4620	7213	<b>7851</b>	5483	5066
R<int>[16]	16	4418	1456	2272	4836	45 676	11 264	43 581	51 441	<b>71 941</b>	56 657	68 311
	32	8853	2917	4448	2468	22 219	5596	20 123	24 745	35 200	27 822	<b>46 537</b>
	64	17 621	5823	8800	1239	11 040	2723	10 722	12 111	17 787	13 811	<b>29 324</b>
	128	35 303	11 657	17 504	611	5393	1431	5126	5717	8675	6953	<b>16 140</b>
	256	70 620	23 311	34 912	312	1888	697	2236	2787	4386	3499	<b>8503</b>
	512	141 125	46 581	69 728	153	778	325	1060	1335	2187	1732	<b>4382</b>
R<double>[2]	16	984	450	1504	4929	108 094	53 704	113 163	91 654	<b>250 680</b>	171 245	77 323
	32	1975	898	2912	2480	58 975	27 004	52 799	46 017	<b>124 924</b>	88 133	50 818
	64	3940	1794	5728	1250	26 607	12 998	27 642	23 430	<b>63 033</b>	43 966	30 609
	128	7863	3587	11 360	615	14 801	6766	13 983	11 607	<b>31 718</b>	21 992	16 934
	256	15 745	7171	22 624	313	6586	3348	6942	5767	<b>15 632</b>	10 999	8918
	512	31 470	14 339	45 152	152	3253	1720	3325	2849	<b>7930</b>	5526	4882
R<double>[16]	16	6547	2242	3296	4764	19 113	9003	21 329	13 449	<b>72 139</b>	56 507	65 271
	32	13 120	4482	6496	2430	9738	4817	10 496	6748	35 394	28 034	<b>41 929</b>
	64	26 144	8962	12 896	1201	4476	2098	5417	3347	17 990	13 987	<b>26 574</b>
	128	52 247	17 923	25 696	601	2171	1159	2574	1644	9247	7068	<b>15 228</b>
	256	104 501	35 843	51 296	299	828	576	1340	819	4615	3536	<b>7852</b>
	512	209 022	71 683	102 496	153	379	276	571	404	2316	1743	<b>4035</b>
R<Str[32]>[2]	16	1489	1250	3168	4827	114 336	53 173	105 648	113 564	<b>126 688</b>	96 718	50 720
	32	2977	2498	6240	2394	63 266	26 620	51 472	57 393	<b>64 721</b>	48 029	29 220
	64	5953	4994	12 384	1150	30 896	12 351	25 692	29 005	<b>31 855</b>	24 668	17 478
	128	11 905	9987	24 672	590	15 587	6267	13 594	14 535	<b>16 089</b>	12 127	9862
	256	23 809	19 971	49 248	299	<b>8162</b>	3183	6761	7321	8064	6002	5019
	512	47 617	39 939	98 400	143	<b>4106</b>	1527	3658	3039	4066	3017	2610
R<Str[32]>[16]	16	10 561	8642	16 608	3680	<b>21 283</b>	8178	18 088	16 871	21 068	16 813	16 012
	32	21 121	17 282	33 120	1829	10 232	4124	9511	8564	<b>10 606</b>	8514	8466
	64	42 241	34 562	66 144	900	<b>5364</b>	2265	4785	4268	5240	4253	4553
	128	84 481	69 123	132 192	458	1598	1009	2196	2162	<b>2648</b>	2134	2333
	256	168 961	138 243	264 288	227	749	470	1060	1075	<b>1320</b>	966	1188
	512	337 921	276 483	528 536	110	374	233	528	529	<b>659</b>	426	582
Client	16	43 860	26 062	68 128	2167	3179	1884	4145	5229	<b>7260</b>	4574	3523
	32	78 424	46 621	139 184	1152	1249	604	2154	2294	<b>3078</b>	2292	1531
	64	154 518	98 438	272 592	540	466	358	906	1361	<b>1444</b>	964	893
	128	318 981	185 362	539 880	248	247	177	421	653	<b>768</b>	461	443
	256	676 492	381 343	1 082 504	126	123	87	221	315	<b>388</b>	216	144
	512	1 290 938	753 998	2 141 312	67	62	45	57	156	<b>204</b>	97	55

**Table B.7**Deserialization throughput for arrays of primitive types with length  $n$  ( $\text{int}[n]$  and  $\text{double}[n]$ ).

Object kind	java [2]	jackson [3]	gson [4]	dsljson [5]	fastjson2 [6]	Kryo [1]		Graal DOSS
						def	ref	
$\text{int}[64]$	37 618	251 389	172 243	197 461	301 280	<b>617 634</b>	436 596	187 658
$\text{int}[128]$	27 337	129 786	80 699	95 099	152 594	<b>322 108</b>	218 517	173 765
$\text{int}[256]$	14 339	66 919	41 272	49 478	79 613	<b>159 025</b>	114 955	157 818
$\text{int}[512]$	7408	33 284	20 879	25 868	41 588	85 035	56 679	<b>167 273</b>
$\text{int}[1024]$	4194	16 671	10 275	12 438	19 470	42 908	30 451	<b>147 903</b>
$\text{int}[2048]$	1663	8439	5027	5973	8841	21 527	15 177	<b>153 234</b>
$\text{int}[4096]$	724	4128	2553	2977	4074	10 673	7534	<b>139 654</b>
$\text{double}[64]$	32 048	39 766	41 167	125 662	127 046	<b>665 010</b>	428 417	133 265
$\text{double}[128]$	23 932	18 366	19 276	61 390	62 689	<b>308 502</b>	222 255	149 701
$\text{double}[256]$	8250	9735	9056	32 490	29 692	<b>174 801</b>	116 688	139 459
$\text{double}[512]$	7225	4625	4661	16 681	16 055	77 095	57 699	<b>137 014</b>
$\text{double}[1024]$	2745	2348	2351	7872	7683	44 229	28 802	<b>134 839</b>
$\text{double}[2048]$	1932	1154	1122	4207	3735	22 544	14 885	<b>130 185</b>
$\text{double}[4096]$	784	568	579	1797	1904	10 980	7472	<b>124 579</b>

**Table B.8**Deserialization throughput for strings of length  $n$  ( $\text{Str}[n]$ ), records with  $n$  primitive and string fields ( $\text{R}<\text{int}>[n]$ ,  $\text{R}<\text{double}>[n]$ ,  $\text{R}<\text{Str}[32]>[n]$ ), and the Client POJO (Client).

Object kind	java [2]	jackson [3]	gson [4]	dsljson [5]	fastjson2 [6]	Kryo [1]		Graal DOSS
						def	ref	
$\text{Str}[32]$	1 872 283	3 395 379	1 783 320	5 711 572	5 623 927	<b>10 484 551</b>	6 744 406	142 806
$\text{Str}[64]$	1 225 173	2 772 522	1 586 876	4 427 345	4 211 187	<b>5 318 256</b>	4 210 199	136 620
$\text{Str}[128]$	893 498	2 014 139	1 238 484	3 021 643	3 025 440	<b>4 432 917</b>	3 389 994	129 365
$\text{Str}[256]$	550 360	1 520 298	835 444	1 901 453	2 091 338	<b>2 383 228</b>	2 143 596	125 483
$\text{Str}[512]$	324 940	880 758	559 229	1 048 605	1 083 752	<b>1 390 559</b>	1 215 616	123 382
$\text{R}<\text{int}>[2]$	64 133	1 596 469	1 108 177	1 923 354	2 947 283	<b>7 022 272</b>	4 109 568	122 866
$\text{R}<\text{int}>[4]$	58 411	1 097 617	741 262	1 120 396	2 042 536	<b>5 407 925</b>	3 282 370	122 866
$\text{R}<\text{int}>[8]$	37 267	749 483	515 023	649 814	1 328 728	<b>3 485 859</b>	2 316 106	126 549
$\text{R}<\text{int}>[16]$	32 762	404 625	326 185	315 811	814 840	<b>2 125 323</b>	1 441 567	130 116
$\text{R}<\text{double}>[2]$	65 040	729 969	530 718	931 167	1 217 279	<b>6 539 167</b>	4 125 750	131 838
$\text{R}<\text{double}>[4]$	50 376	382 239	379 623	468 674	674 288	<b>5 096 519</b>	3 130 102	129 743
$\text{R}<\text{double}>[8]$	45 633	239 340	225 561	255 494	415 544	<b>3 690 160</b>	2 247 270	116 525
$\text{R}<\text{double}>[16]$	35 264	124 674	122 359	146 674	230 628	<b>1 979 432</b>	1 340 712	112 843
$\text{R}<\text{Str}[32]>[2]$	52 554	1 275 606	904 955	1 916 064	2 556 405	<b>3 930 739</b>	2 477 765	121 600
$\text{R}<\text{Str}[32]>[4]$	39 203	954 692	650 438	1 037 638	1 774 669	<b>2 356 955</b>	1 490 479	123 341
$\text{R}<\text{Str}[32]>[8]$	34 090	588 885	444 548	606 663	1 060 371	<b>1 424 540</b>	1 036 089	127 946
$\text{R}<\text{Str}[32]>[16]$	24 652	316 717	267 182	296 766	557 797	<b>766 168</b>	547 654	100 702
Client	5043	14 461	12 999	52 885	77 629	<b>153 808</b>	132 619	136 255



Table B.9

Deserialization throughput for lists of variable types and sizes.

Type	Size	java [2]	jackson [3]	gson [4]	dsljson [5]	fastjson2 [6]	Kryo [1] def	ref	Graal DOSS
int	16	63 543	733 459	491 930	570 570	1 003 186	<b>2 336 265</b>	1 550 977	123 205
	32	44 667	394 429	303 044	296 722	572 248	<b>1 355 123</b>	940 566	121 261
	64	29 396	216 936	157 534	176 853	301 176	<b>719 201</b>	442 639	120 434
	128	22 962	116 134	83 157	87 973	142 653	<b>360 335</b>	242 924	117 061
	256	12 372	57 842	42 277	43 434	76 602	<b>192 718</b>	133 845	137 530
	512	6574	29 171	21 068	22 489	42 218	87 097	65 342	<b>126 211</b>
double	16	58 323	135 116	143 919	373 050	437 126	<b>2 428 473</b>	1 494 059	125 111
	32	45 960	74 388	73 539	220 514	203 700	1 291 844	874 392	137 879
	64	33 996	40 019	38 509	115 841	105 418	<b>716 957</b>	403 663	105 711
	128	18 292	19 694	19 361	58 965	57 224	<b>394 596</b>	248 336	120 885
	256	10 653	9940	9517	30 345	30 136	<b>185 807</b>	122 217	132 871
	512	5989	4386	4739	15 620	14 292	101 480	56 966	<b>129 397</b>
Str[32]	16	69 094	444 789	353 922	561 950	706 063	<b>812 142</b>	595 884	101 744
	32	57 202	237 612	215 359	293 955	350 053	<b>463 531</b>	314 910	127 058
	64	32 995	129 037	116 386	175 020	212 555	<b>233 012</b>	164 749	124 749
	128	19 420	62 101	60 386	77 470	106 831	121 343	89 406	<b>126 986</b>
	256	6445	32 425	32 676	40 600	55 845	61 174	43 233	<b>123 936</b>
	512	2957	17 020	16 931	21 024	28 116	29 953	22 135	<b>124 477</b>
R<int>[2]	16	23 048	202 827	152 337	181 513	254 403	<b>640 752</b>	417 594	107 371
	32	14 823	110 846	78 023	92 599	129 163	<b>339 555</b>	196 390	118 270
	64	8445	56 734	40 865	49 222	64 824	<b>183 930</b>	115 251	115 528
	128	4808	26 572	20 393	24 855	32 607	95 781	59 224	<b>115 665</b>
	256	2569	14 488	10 716	12 313	16 253	47 544	30 301	<b>140 177</b>
	512	1386	6963	5262	5663	7826	23 320	15 397	<b>135 637</b>
R<int>[16]	16	6656	37 281	19 506	26 552	36 593	<b>165 260</b>	106 084	114 803
	32	3166	18 729	9765	14 653	17 626	70 220	54 640	<b>141 795</b>
	64	1339	9237	4199	7126	8565	40 575	27 718	<b>127 626</b>
	128	656	4319	2339	3396	4216	17 713	14 111	<b>120 450</b>
	256	515	2076	1226	1518	2099	8869	7035	<b>140 981</b>
	512	265	1023	599	706	1062	5037	3508	<b>133 370</b>
R<double>[2]	16	22 065	59 940	55 511	136 327	149 901	<b>635 866</b>	391 664	110 276
	32	13 638	26 982	28 756	64 872	69 227	<b>347 520</b>	208 637	116 408
	64	7082	14 814	13 903	37 633	34 661	<b>181 307</b>	110 097	104 597
	128	4123	7573	7116	19 021	18 740	90 792	57 103	<b>127 205</b>
	256	2583	3659	3450	9237	9002	46 929	28 249	<b>128 898</b>
	512	1275	1668	1745	4283	4261	23 577	14 229	<b>132 578</b>
R<double>[16]	16	4819	7434	6958	18 545	18 512	<b>164 297</b>	103 472	109 652
	32	2182	3903	3326	10 726	9374	85 090	53 639	<b>132 478</b>
	64	1190	1962	1632	5228	4092	41 877	27 131	<b>101 741</b>
	128	670	948	799	2549	2001	18 374	13 807	<b>122 009</b>
	256	483	463	405	1139	992	11 338	7114	<b>112 534</b>
	512	227	232	201	581	562	5837	3515	<b>123 746</b>
R<Str[32]>[2]	16	16 683	148 800	110 609	185 022	231 669	<b>272 033</b>	222 455	124 939
	32	8933	71 381	67 292	96 973	122 425	<b>162 994</b>	115 738	132 585
	64	5104	41 275	35 373	51 701	61 195	85 183	59 341	<b>130 968</b>
	128	2560	20 301	18 957	25 910	26 271	41 484	30 165	<b>135 227</b>
	256	1227	10 600	7406	14 417	15 213	21 621	15 361	<b>129 841</b>
	512	478	5490	4618	6927	6853	10 908	6519	<b>127 971</b>
R<Str[32]>[16]	16	2098	23 169	16 620	28 799	30 974	54 866	39 584	<b>128 563</b>
	32	1563	11 528	8577	15 595	15 324	26 980	19 948	<b>130 283</b>
	64	616	5929	4252	7888	7235	13 678	9833	<b>123 003</b>
	128	393	3241	2137	2639	3175	6787	5011	<b>101 941</b>
	256	117	1647	1040	1447	1886	3386	2445	<b>91 439</b>
	512	92	771	414	624	913	1670	1190	<b>92 197</b>
Client	16	714	4415	3548	5449	5265	10 683	5795	<b>118 936</b>
	32	360	1889	1954	2138	2121	4959	3272	<b>111 801</b>
	64	151	1058	866	1177	1064	2087	1683	<b>97 880</b>
	128	102	521	427	547	514	1227	880	<b>82 802</b>
	256	53	275	224	284	266	632	418	<b>79 725</b>
	512	30	131	97	127	104	272	193	<b>78 140</b>



Table B.10

Deserialization throughput for maps that have integer keys.

Map (Key type: int)		java	jackson	gson	dsljson	fastjson2	Kryo [1]		Graal
Value type	Size	[2]	[3]	[4]	[5]	[6]	def	ref	DOSS
int	16	48 176	377 743	284 141	335 981	421 008	<b>814 154</b>	738 622	110 533
	32	34 372	213 398	161 757	180 221	215 293	<b>426 125</b>	385 086	124 924
	64	23 567	113 942	84 679	92 515	108 800	<b>276 257</b>	202 597	114 604
	128	10 202	60 079	44 197	45 455	55 781	<b>141 703</b>	100 302	122 797
	256	6606	31 357	21 475	22 826	26 109	69 458	50 834	<b>115 173</b>
	512	3268	14 792	10 426	11 496	13 080	28 684	23 218	<b>106 296</b>
Str[8]	16	44 440	389 399	362 258	385 504	440 400	<b>663 566</b>	463 145	115 231
	32	32 083	214 756	192 384	202 503	231 949	<b>347 166</b>	247 379	118 859
	64	18 927	123 815	101 013	109 355	117 919	<b>175 637</b>	126 145	124 669
	128	10 353	63 597	49 818	54 564	54 573	97 385	66 320	<b>121 085</b>
	256	4301	30 006	26 862	27 914	27 865	45 714	31 508	<b>123 498</b>
	512	2918	14 153	13 244	11 789	13 760	24 653	16 729	<b>109 894</b>
Str[32]	16	33 918	299 765	237 492	310 991	364 675	<b>510 597</b>	371 010	115 842
	32	25 147	163 430	136 857	131 653	183 012	<b>263 483</b>	200 999	119 547
	64	18 224	82 772	73 279	85 049	93 689	<b>140 741</b>	102 447	114 099
	128	8993	42 379	37 575	42 488	35 933	66 689	50 801	<b>120 797</b>
	256	3388	20 135	17 817	21 063	21 476	35 705	25 526	<b>116 820</b>
	512	1046	9714	8969	11 010	10 730	17 863	13 601	<b>110 696</b>
R<int>[2]	16	18 203	151 250	114 102	176 201	195 310	<b>453 507</b>	300 201	106 234
	32	11 018	83 291	64 332	90 360	99 380	<b>237 951</b>	157 061	107 383
	64	5916	45 550	31 569	46 243	48 621	116 812	83 078	<b>123 878</b>
	128	4133	22 687	16 231	22 977	24 330	59 625	40 827	<b>115 350</b>
	256	2036	10 738	8042	12 054	11 660	30 272	21 108	<b>115 547</b>
	512	1077	5280	3941	5644	5412	15 004	10 433	<b>115 181</b>
R<int>[16]	16	5635	33 131	19 026	31 652	35 757	<b>143 774</b>	98 041	120 826
	32	3013	17 495	9352	15 464	17 800	74 023	50 169	108 989
	64	1373	8178	4668	7651	8527	37 280	25 220	<b>104 517</b>
	128	797	3883	2371	3641	4067	13 674	12 441	<b>104 223</b>
	256	455	1893	1137	1536	1968	9335	6389	<b>114 394</b>
	512	253	949	554	804	991	4449	3060	<b>119 523</b>
R<double>[2]	16	14 983	51 619	49 789	128 794	115 430	<b>425 025</b>	279 772	125 202
	32	9547	27 081	24 276	68 737	56 435	<b>219 120</b>	148 085	121 056
	64	6350	13 674	13 177	33 955	32 829	<b>113 279</b>	76 250	110 717
	128	3962	6705	6305	17 540	14 332	60 183	39 672	<b>103 001</b>
	256	1602	3410	3140	8656	7275	30 342	20 401	<b>102 441</b>
	512	981	1616	1493	4357	3524	14 625	9997	<b>109 946</b>
R<double>[16]	16	4711	7773	6569	21 666	18 517	<b>140 847</b>	92 543	127 650
	32	2410	3636	3329	11 259	9272	74 921	46 746	<b>107 365</b>
	64	1363	1888	1580	4261	4764	37 066	24 112	<b>107 769</b>
	128	546	912	806	2734	2233	18 851	12 655	<b>111 181</b>
	256	486	435	389	1201	1093	9411	6107	<b>116 818</b>
	512	253	224	197	602	532	4638	3031	<b>121 746</b>
R<Str[32]>[2]	16	14 248	115 453	112 701	158 044	162 899	<b>247 413</b>	172 765	124 491
	32	6456	62 574	55 209	80 019	80 339	94 213	87 872	<b>127 710</b>
	64	4280	31 071	29 076	42 267	40 618	66 816	46 679	<b>113 063</b>
	128	2411	15 758	14 766	21 464	21 342	33 442	23 240	<b>116 800</b>
	256	1073	8026	7243	10 740	10 166	16 831	11 416	<b>102 717</b>
	512	582	4187	3535	5345	4959	6235	5808	<b>110 904</b>
R<Str[32]>[16]	16	3363	21 295	15 837	28 484	27 993	49 645	36 114	<b>120 659</b>
	32	1383	11 363	8280	14 334	13 952	25 464	18 084	<b>113 688</b>
	64	663	5731	4092	7750	7296	12 782	8978	<b>116 276</b>
	128	410	2859	2055	2886	3487	6099	4497	<b>101 791</b>
	256	208	1455	982	1506	1719	3147	2218	<b>87 404</b>
	512	110	736	515	711	797	1555	1106	<b>86 673</b>
Client	16	674	4620	3778	5670	4314	9226	6942	<b>99 180</b>
	32	272	2002	1512	2170	2229	4654	3411	<b>95 990</b>
	64	207	986	883	1077	1098	2358	1749	<b>94 508</b>
	128	114	519	407	537	493	1148	903	<b>86 272</b>
	256	54	249	201	264	184	589	405	<b>77 296</b>
	512	31	122	99	92	121	284	192	<b>70 614</b>

Table B.11

Deserialization throughput for maps that have strings of length 8 as keys.

Map (Key type: int)		java	jackson	gson	dsljson	fastjson2	Kryo [1]		Graal
Value type	Size	[2]	[3]	[4]	[5]	[6]	def	ref	DOSS
int	16	36 336	398 158	282 571	410 973	424 300	<b>644 522</b>	444 012	119 826
	32	32 986	211 345	161 563	209 724	218 264	<b>328 591</b>	237 010	112 769
	64	18 776	115 193	82 095	113 334	115 633	<b>171 014</b>	123 639	110 649
	128	12 753	57 873	40 646	56 405	56 392	91 907	63 755	<b>111 384</b>
	256	5060	30 916	20 786	26 315	27 778	42 862	30 398	<b>115 362</b>
	512	2415	15 752	10 690	12 606	12 836	22 502	12 106	<b>110 228</b>
Str[8]	16	78 122	371 029	333 760	<b>473 723</b>	461 902	588 928	371 083	117 120
	32	30 205	209 410	210 156	222 099	<b>238 855</b>	202 262	198 515	113 373
	64	31 143	114 160	110 072	125 379	126 012	<b>145 706</b>	98 017	109 744
	128	11 609	58 010	57 869	59 122	61 227	74 374	51 648	<b>110 324</b>
	256	5250	28 007	29 213	30 861	27 675	38 424	24 692	<b>110 781</b>
	512	2915	14 675	13 731	16 887	13 265	19 067	13 109	<b>110 160</b>
Str[32]	16	39 135	300 370	257 497	371 105	384 129	<b>458 113</b>	320 638	118 881
	32	43 894	142 676	142 992	177 965	195 486	<b>235 891</b>	166 319	105 768
	64	25 197	77 776	72 212	95 205	95 172	<b>122 060</b>	86 690	107 028
	128	10 933	39 113	35 395	48 291	48 178	61 222	42 731	<b>123 335</b>
	256	3893	19 551	19 222	24 275	22 941	31 380	21 396	<b>112 798</b>
	512	885	9732	9262	12 315	11 091	15 335	11 224	<b>103 888</b>
R<int>[2]	16	16 981	170 067	119 965	165 436	195 383	<b>343 355</b>	164 604	114 081
	32	9352	88 735	61 783	89 861	100 855	<b>185 549</b>	121 171	117 802
	64	6938	41 045	32 333	45 053	51 827	95 412	65 013	<b>117 685</b>
	128	3949	22 446	16 220	22 157	24 461	48 509	22 789	<b>120 945</b>
	256	1991	10 862	7991	11 042	11 465	24 526	16 420	<b>113 555</b>
	512	1058	5404	3951	5632	5557	12 775	8454	<b>114 960</b>
R<int>[16]	16	6102	33 361	18 686	30 413	36 250	<b>132 431</b>	90 677	117 207
	32	2383	17 566	9457	15 855	17 677	67 682	46 796	<b>109 509</b>
	64	1649	7857	4412	7678	8509	34 010	23 519	<b>106 041</b>
	128	882	3722	2333	3712	3808	16 644	11 337	<b>111 909</b>
	256	455	1973	1170	1508	1943	8359	5765	<b>97 948</b>
	512	244	992	571	808	983	3995	2811	<b>101 882</b>
R<double>[2]	16	18 907	53 462	49 622	127 764	136 093	<b>334 287</b>	229 423	114 000
	32	10 088	27 171	27 031	66 416	60 500	<b>179 825</b>	120 608	110 243
	64	6912	12 930	12 961	36 670	31 365	96 215	63 474	<b>103 133</b>
	128	4067	6999	6318	17 379	15 364	46 539	31 874	<b>116 387</b>
	256	1825	3232	3112	8712	7281	24 727	16 341	<b>99 343</b>
	512	903	1647	1548	4477	3616	12 822	8417	<b>105 721</b>
R<double>[16]	16	4759	7136	6752	22 466	18 269	<b>136 020</b>	85 580	120 824
	32	2967	3444	3275	11 218	9200	69 322	43 933	<b>123 015</b>
	64	1394	1780	1610	5503	4780	36 509	22 805	<b>118 286</b>
	128	502	909	789	2759	2212	17 556	10 897	<b>119 643</b>
	256	432	454	398	1211	1131	8957	5816	<b>111 162</b>
	512	259	223	191	625	563	4420	2899	<b>113 037</b>
R<Str[32]>[2]	16	11 209	113 681	110 279	166 588	164 009	<b>235 087</b>	156 057	119 402
	32	10 193	59 790	58 515	85 660	86 141	<b>115 155</b>	80 650	100 468
	64	5846	32 058	31 272	46 346	41 250	63 315	42 969	<b>105 559</b>
	128	2194	17 172	15 834	22 968	20 013	30 947	21 420	<b>103 920</b>
	256	984	8566	7975	8093	10 304	14 851	10 811	<b>110 692</b>
	512	598	4346	3976	5968	5170	7546	5286	<b>95 699</b>
R<Str[32]>[16]	16	2989	22 660	16 397	27 744	27 634	49 038	35 326	<b>102 180</b>
	32	1534	10 768	8313	15 668	13 922	25 479	17 841	<b>110 535</b>
	64	687	5509	4296	7881	7294	12 289	8689	<b>110 234</b>
	128	240	2817	2115	2754	3502	6243	4432	<b>97 991</b>
	256	202	1529	1061	1428	1773	3063	2167	<b>80 352</b>
	512	108	748	543	711	844	1547	1092	<b>75 980</b>
Client	16	678	4188	3125	4654	4296	11 015	6396	<b>101 696</b>
	32	357	2401	1687	1961	2034	4655	3226	<b>95 073</b>
	64	217	1026	759	1109	1002	2539	1091	<b>88 477</b>
	128	74	500	396	596	538	1184	876	<b>68 037</b>
	256	62	254	196	276	248	586	426	<b>69 012</b>
	512	31	124	93	146	127	273	192	<b>65 686</b>

**Table B.12**

S/D frameworks used in microbenchmarks.

Framework	Version	Release date
jackson-databind [3]	2.17.0	Mar 2024
gson [4]	2.11.0	May 2024
dsljson [5]	2.0.2	Aug 2023
fastjson2 [6]	2.0.31	May 2023
Kryo [1]	5.6.0	Jan 2024

## References

- [1] EsotericSoftware, EsotericSoftware/Kryo: Java binary serialization and cloning: Fast, efficient, automatic (2008).  
URL <https://github.com/EsotericSoftware/kryo>
- [2] Oracle, Java Object Serialization Specification, online.  
URL <https://docs.oracle.com/en/java/javase/17/docs/specs/serialization/serial-arch.html>
- [3] FasterXML, Jackson, online.  
URL <https://github.com/FasterXML/jackson>
- [4] Google, gson, online (2008).  
URL <https://github.com/google/gson>
- [5] New Generation Software, dsl-json, online (2023).  
URL <https://github.com/ngs-doo/dsl-json>
- [6] Alibaba, fastjson 2.0, online (2024).  
URL <https://github.com/alibaba/fastjson>