

GIN parallel builds results

hardware

i5 - 4 cores, 16GB RAM, 6x100GB SATA SSD in RAID0
xeon - 16/32 cores, 64GB RAM, 1TB NVMe SSD

dataset

messages table has 1.6M rows with archives from pgsqll mailing lists

```
test=# \d messages
```

Column	Type	Collation	Nullable	Default
list_name	text			
file_name	text			
file_index	integer			
msg_hash	text			
msg_raw	text			
msg_date	timestamp with time zone			
msg_from	text			
msg_subject	text			
msg_id	text			
msg_thread_id	text			
msg_headers	jsonb			
msg_body	text			
msg_subject_tsvector	tsvector			
msg_body_tsvector	tsvector			

```
test=# \d+
```

Schema	Name	Type	Owner	Persistence	Access method	Size
public	messages	table	tomas	permanent	heap	22 GB

(3 rows)

```
create index trgm on messages using gin (msg_body gin_trgm_ops);  
create index tsvector on messages using gin (msg_body_tsvector);  
create index jsonb on messages using gin (msg_headers);  
create index jsonb_hash on messages using gin (msg_headers jsonb_path_ops);
```

test=# \di+

List of relations

Schema	Name	Type	Owner	Table	Persistence	Access method	Size
public	jsonb	index	tomas	messages	permanent	gin	1340 MB
public	jsonb_hash	index	tomas	messages	permanent	gin	658 MB
public	trgm	index	tomas	messages	permanent	gin	3264 MB
public	tsvector	index	tomas	messages	permanent	gin	3236 MB

(4 rows)

patches

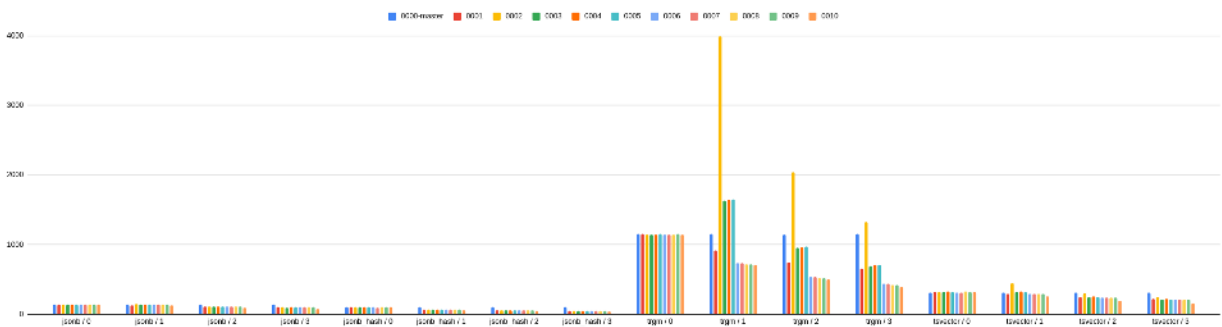
- 0001-Allow-parallel-create-for-GIN-indexes.patch
- 0002-Use-mergesort-in-the-leader-process.patch
- 0003-Remove-the-explicit-pg_qsort-in-workers.patch
- 0004-Compress-TID-lists-before-writing-tuples-t.patch
- 0005-Collect-and-print-compression-stats.patch
- 0006-Enforce-memory-limit-when-combining-tuples.patch
- 0007-Detect-wrap-around-in-parallel-callback.patch
- 0008-Use-a-single-GIN-tuplesort.patch
- 0009-Reduce-the-size-of-GinTuple-by-12-bytes.patch
- 0010-WIP-parallel-inserts-into-GIN-index.patch

i5 / work_mem = 4MB

absolute timings

Timing in seconds for different indexes on the messages table, using a different number of parallel workers (0 - serial build). With different patches of the series applied (0000 - no patches).

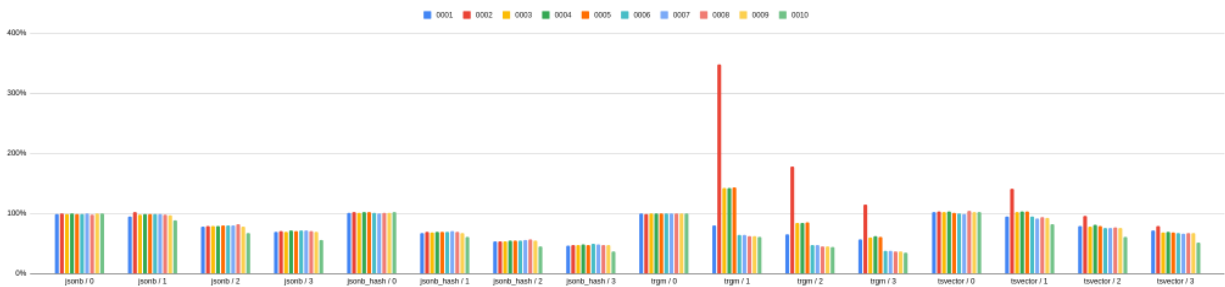
label	0000-master	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	142	141	142	141	142	142	141	142	140	142	142
jsonb / 1	142	134	145	140	140	141	141	140	139	138	126
jsonb / 2	142	110	112	112	112	113	114	114	116	110	95
jsonb / 3	142	99	100	98	102	101	101	102	99	99	79
jsonb_hash / 0	99	100	101	100	100	101	100	99	100	100	101
jsonb_hash / 1	99	67	69	68	69	69	69	70	69	66	61
jsonb_hash / 2	100	54	54	54	55	54	55	56	57	55	45
jsonb_hash / 3	101	47	48	48	48	48	49	49	48	48	38
trgm / 0	1149	1148	1143	1145	1146	1153	1145	1146	1145	1154	1147
trgm / 1	1149	919	3992	1630	1637	1648	738	738	719	719	705
trgm / 2	1143	748	2037	957	965	972	537	540	521	522	500
trgm / 3	1147	651	1320	690	707	706	440	438	423	421	396
tsvector / 0	314	320	322	319	325	318	315	311	326	321	321
tsvector / 1	314	296	443	320	325	324	297	289	294	292	259
tsvector / 2	314	249	301	245	254	249	238	238	242	237	193
tsvector / 3	312	225	247	212	217	215	210	207	211	211	161



relative timings

Timing relative to 0000-master, i.e. always serial build. Ideal value (perfect parallel speedup) would be $1 / (1 + \text{workers})$, but that is not really achievable.

label	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	99%	100%	99%	100%	99%	99%	100%	98%	100%	100%
jsonb / 1	94%	102%	98%	99%	99%	99%	99%	98%	97%	88%
jsonb / 2	78%	79%	79%	79%	79%	80%	80%	82%	78%	67%
jsonb / 3	70%	70%	69%	72%	71%	72%	72%	70%	70%	56%
jsonb_hash / 0	101%	102%	101%	102%	102%	101%	100%	101%	101%	102%
jsonb_hash / 1	67%	70%	68%	69%	69%	69%	70%	69%	67%	62%
jsonb_hash / 2	54%	53%	54%	55%	54%	55%	56%	56%	55%	45%
jsonb_hash / 3	46%	47%	47%	48%	48%	49%	48%	48%	47%	37%
trgm / 0	100%	99%	100%	100%	100%	100%	100%	100%	100%	100%
trgm / 1	80%	347%	142%	142%	143%	64%	64%	63%	63%	61%
trgm / 2	65%	178%	84%	84%	85%	47%	47%	46%	46%	44%
trgm / 3	57%	115%	60%	62%	62%	38%	38%	37%	37%	35%
tsvector / 0	102%	103%	102%	104%	101%	100%	99%	104%	102%	102%
tsvector / 1	94%	141%	102%	104%	103%	95%	92%	94%	93%	82%
tsvector / 2	79%	96%	78%	81%	79%	76%	76%	77%	75%	62%
tsvector / 3	72%	79%	68%	69%	69%	67%	66%	68%	67%	51%

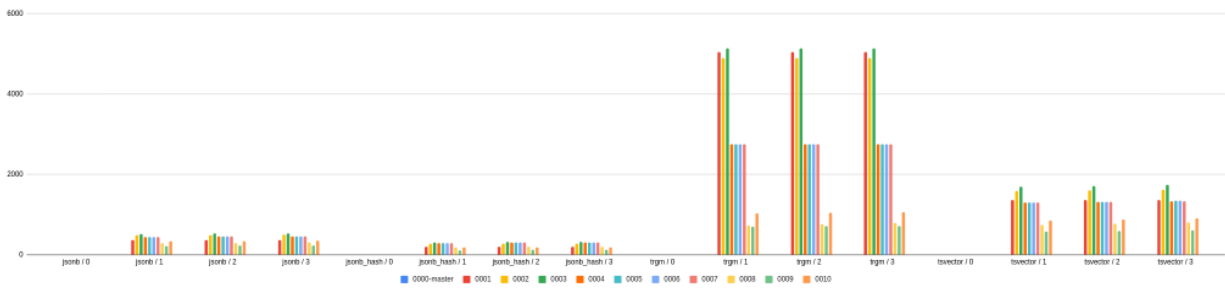


temporary space

Amount of temporary data generated by the different builds (in MB). The 0000-master never generates any temporary files (same as serial builds with 0 workers).

label	0000-master	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	0	0	0	0	0	0	0	0	0	0	0
jsonb / 1	0	365	486	521	446	446	446	446	287	222	333

jsonb / 2	0	367	494	530	456	456	456	456	295	228	342
jsonb / 3	0	368	500	536	463	462	463	463	300	232	349
jsonb_hash / 0	0	0	0	0	0	0	0	0	0	0	0
jsonb_hash / 1	0	206	275	312	293	293	293	293	190	117	179
jsonb_hash / 2	0	208	280	318	300	300	300	300	195	120	184
jsonb_hash / 3	0	208	284	322	305	304	305	304	199	122	187
trgm / 0	0	0	0	0	0	0	0	0	0	0	0
trgm / 1	0	5042	4893	5136	2748	2747	2749	2749	729	694	1026
trgm / 2	0	5044	4894	5138	2750	2750	2752	2752	761	707	1041
trgm / 3	0	5045	4896	5139	2752	2752	2753	2754	793	722	1056
tsvector / 0	0	0	0	0	0	0	0	0	0	0	0
tsvector / 1	0	1360	1584	1694	1297	1297	1299	1299	750	578	855
tsvector / 2	0	1360	1607	1717	1326	1325	1325	1325	781	594	885
tsvector / 3	0	1363	1622	1738	1341	1344	1344	1341	800	611	905

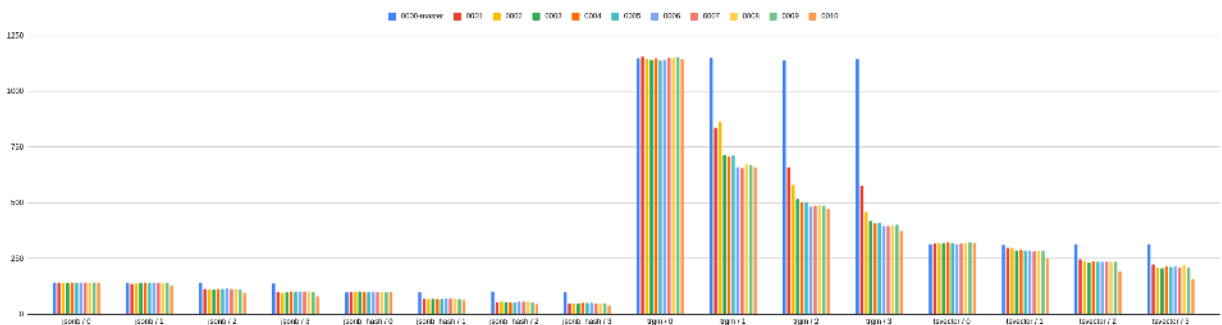


i5 / work_mem = 32MB

absolute timings

Timing in seconds for different indexes on the messages table, using a different number of parallel workers (0 - serial build). With different patches of the series applied (0000 - no patches).

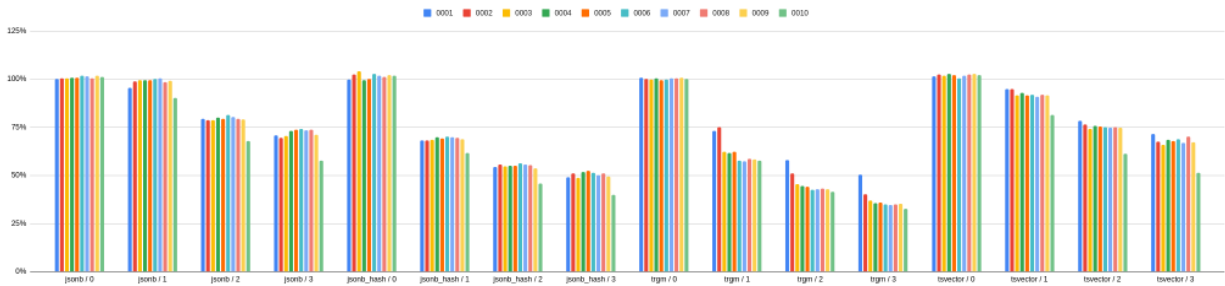
label	0000-master	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	141	141	142	142	142	142	144	143	142	144	143
jsonb / 1	142	136	141	141	141	141	143	143	140	141	128
jsonb / 2	142	113	112	112	114	113	116	114	113	113	97
jsonb / 3	141	100	98	99	103	103	104	103	104	100	81
jsonb_hash / 0	99	99	102	103	99	99	102	101	101	101	101
jsonb_hash / 1	101	69	69	69	70	70	71	70	70	69	62
jsonb_hash / 2	102	55	57	56	56	56	57	57	57	55	47
jsonb_hash / 3	99	48	50	48	51	52	51	49	50	49	39
trgm / 0	1147	1156	1146	1142	1149	1140	1143	1150	1151	1154	1146
trgm / 1	1150	839	862	715	707	714	661	658	673	669	661
trgm / 2	1139	660	582	517	504	504	483	486	491	486	472
trgm / 3	1144	577	459	420	408	411	397	396	398	402	374
tsvector / 0	314	318	322	320	322	321	315	319	322	322	321
tsvector / 1	313	297	297	286	290	286	287	284	287	286	254
tsvector / 2	314	246	240	233	238	237	236	234	236	234	192
tsvector / 3	314	224	212	206	214	213	215	209	220	211	161



relative timings

Timing relative to 0000-master, i.e. always serial build. Ideal value (perfect parallel speedup) would be $1 / (1 + \text{workers})$, but that is not really achievable.

label	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	100%	100%	100%	101%	101%	102%	101%	100%	102%	101%
jsonb / 1	96%	99%	99%	99%	99%	100%	100%	98%	99%	90%
jsonb / 2	79%	79%	78%	80%	79%	81%	80%	79%	79%	68%
jsonb / 3	71%	69%	70%	73%	74%	74%	73%	74%	71%	58%
jsonb_hash / 0	100%	102%	104%	99%	100%	103%	102%	101%	102%	102%
jsonb_hash / 1	68%	68%	69%	70%	69%	70%	70%	70%	69%	62%
jsonb_hash / 2	54%	55%	55%	55%	55%	56%	56%	55%	54%	46%
jsonb_hash / 3	49%	51%	49%	51%	52%	51%	50%	51%	49%	40%
trgm / 0	101%	100%	100%	100%	99%	100%	100%	100%	101%	100%
trgm / 1	73%	75%	62%	61%	62%	57%	57%	58%	58%	57%
trgm / 2	58%	51%	45%	44%	44%	42%	43%	43%	43%	41%
trgm / 3	50%	40%	37%	36%	36%	35%	35%	35%	35%	33%
tsvector / 0	101%	102%	102%	103%	102%	100%	102%	102%	103%	102%
tsvector / 1	95%	95%	91%	93%	92%	92%	91%	92%	91%	81%
tsvector / 2	78%	76%	74%	76%	75%	75%	75%	75%	75%	61%
tsvector / 3	71%	67%	66%	68%	68%	69%	67%	70%	67%	51%

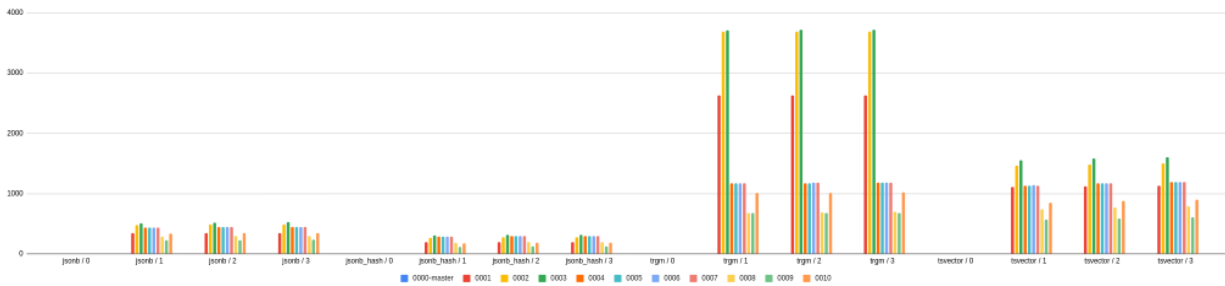


temporary space

Amount of temporary data generated by the different builds (in MB). The 0000-master never generates any temporary files (same as serial builds with 0 workers).

label	0000-master	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	0	0	0	0	0	0	0	0	0	0	0
jsonb / 1	0	341	474	507	432	432	432	432	286	221	333
jsonb / 2	0	344	483	517	443	443	443	443	295	227	342
jsonb / 3	0	345	488	523	450	450	450	449	300	231	348
jsonb_hash / 0	0	0	0	0	0	0	0	0	0	0	0
jsonb_hash / 1	0	195	269	306	286	286	286	286	189	117	178

jsonb_hash / 2	0	196	275	312	293	293	293	293	195	120	184
jsonb_hash / 3	0	197	278	316	297	297	297	297	198	122	187
trgm / 0	0	0	0	0	0	0	0	0	0	0	0
trgm / 1	0	2635	3689	3713	1174	1174	1176	1176	681	673	1006
trgm / 2	0	2635	3689	3714	1176	1175	1177	1177	689	678	1011
trgm / 3	0	2635	3690	3715	1177	1177	1179	1179	696	682	1016
tsvector / 0	0	0	0	0	0	0	0	0	0	0	0
tsvector / 1	0	1112	1459	1549	1134	1133	1140	1135	736	567	846
tsvector / 2	0	1122	1486	1581	1170	1172	1170	1171	767	589	877
tsvector / 3	0	1129	1505	1602	1194	1193	1194	1194	792	606	898

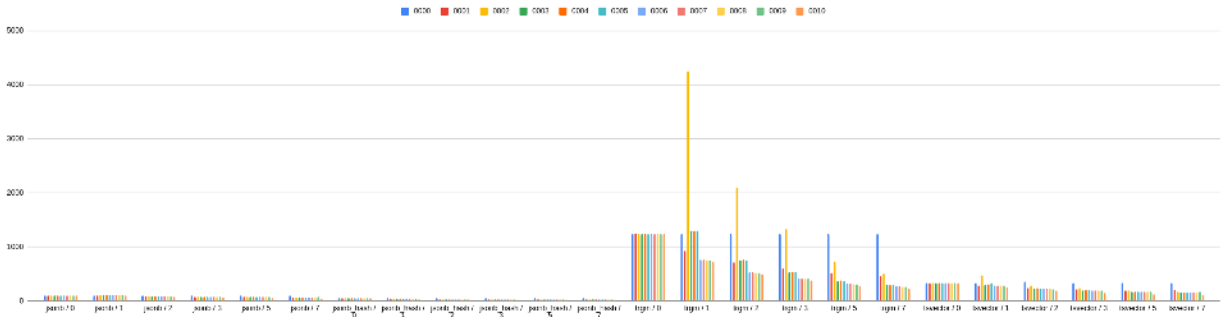


xeon / work_mem = 4MB

absolute timings

Timing in seconds for different indexes on the messages table, using a different number of parallel workers (0 - serial build). With different patches of the series applied (0000 - no patches).

label	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	96	97	97	97	97	98	97	98	97	97	97
jsonb / 1	97	102	112	106	108	108	108	108	107	106	95
jsonb / 2	97	85	87	86	92	87	88	88	86	86	71
jsonb / 3	97	77	77	76	77	79	80	78	77	80	58
jsonb / 5	97	71	70	70	71	71	70	71	70	69	49
jsonb / 7	98	68	66	68	68	68	68	68	67	69	45
jsonb_hash / 0	53	54	53	53	53	54	53	53	53	53	53
jsonb_hash / 1	53	36	37	36	37	38	38	39	38	37	31
jsonb_hash / 2	53	30	30	30	32	32	32	32	32	31	23
jsonb_hash / 3	53	28	27	27	29	29	29	29	28	28	19
jsonb_hash / 5	53	26	24	25	25	26	26	26	27	26	17
jsonb_hash / 7	53	25	23	24	24	25	25	25	25	26	14
trgm / 0	1239	1250	1238	1237	1239	1233	1235	1237	1238	1233	1235
trgm / 1	1238	923	4251	1293	1293	1289	759	757	746	744	729
trgm / 2	1232	709	2094	747	760	752	532	529	519	517	495
trgm / 3	1236	602	1329	529	538	536	415	416	407	407	382
trgm / 5	1234	513	727	363	371	370	316	317	310	311	274
trgm / 7	1236	461	499	293	293	291	268	272	263	263	221
tsvector / 0	327	330	330	330	331	326	331	328	332	330	333
tsvector / 1	327	289	472	300	309	330	285	285	281	278	253
tsvector / 2	354	239	288	226	237	229	224	222	220	219	188
tsvector / 3	329	217	236	193	204	201	196	193	192	191	148
tsvector / 5	329	191	185	169	172	171	172	172	172	170	119
tsvector / 7	330	205	165	158	159	157	160	159	159	162	106

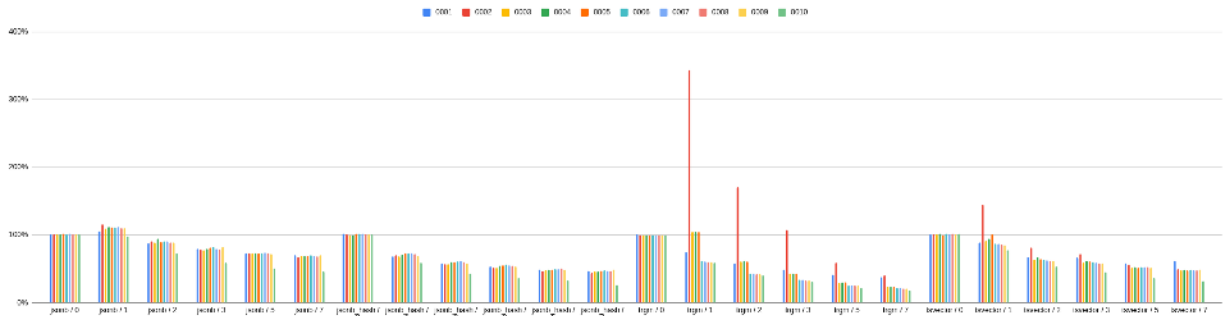


relative timings

Timing relative to 0000-master, i.e. always serial build. Ideal value (perfect parallel speedup) would be $1 / (1 + \text{workers})$, but that is not really achievable.

label	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	101%	101%	101%	100%	102%	101%	101%	100%	101%	101%
jsonb / 1	105%	116%	109%	111%	111%	111%	111%	110%	110%	98%
jsonb / 2	88%	90%	89%	95%	89%	90%	91%	89%	88%	73%
jsonb / 3	79%	79%	78%	79%	81%	82%	80%	79%	82%	59%
jsonb / 5	73%	73%	73%	73%	73%	73%	73%	73%	72%	50%
jsonb / 7	70%	67%	69%	69%	69%	70%	69%	68%	70%	46%
jsonb_hash / 0	102%	100%	100%	100%	102%	101%	101%	101%	101%	101%
jsonb_hash / 1	68%	70%	68%	71%	73%	73%	73%	72%	69%	59%
jsonb_hash / 2	58%	57%	57%	60%	60%	61%	62%	60%	58%	43%
jsonb_hash / 3	54%	52%	52%	54%	55%	56%	55%	54%	53%	36%
jsonb_hash / 5	49%	46%	47%	48%	49%	50%	50%	50%	49%	33%
jsonb_hash / 7	47%	44%	46%	46%	46%	48%	47%	47%	49%	27%
trgm / 0	101%	100%	100%	100%	100%	100%	100%	100%	100%	100%
trgm / 1	75%	343%	104%	104%	104%	61%	61%	60%	60%	59%
trgm / 2	58%	170%	61%	62%	61%	43%	43%	42%	42%	40%
trgm / 3	49%	108%	43%	43%	43%	34%	34%	33%	33%	31%
trgm / 5	42%	59%	29%	30%	30%	26%	26%	25%	25%	22%
trgm / 7	37%	40%	24%	24%	24%	22%	22%	21%	21%	18%
tsvector / 0	101%	101%	101%	101%	100%	101%	100%	102%	101%	102%
tsvector / 1	88%	144%	92%	95%	101%	87%	87%	86%	85%	77%
tsvector / 2	67%	81%	64%	67%	65%	63%	63%	62%	62%	53%
tsvector / 3	66%	72%	59%	62%	61%	60%	59%	58%	58%	45%

tsvector / 5	58%	56%	51%	52%	52%	52%	52%	52%	52%	36%
tsvector / 7	62%	50%	48%	48%	48%	49%	48%	48%	49%	32%

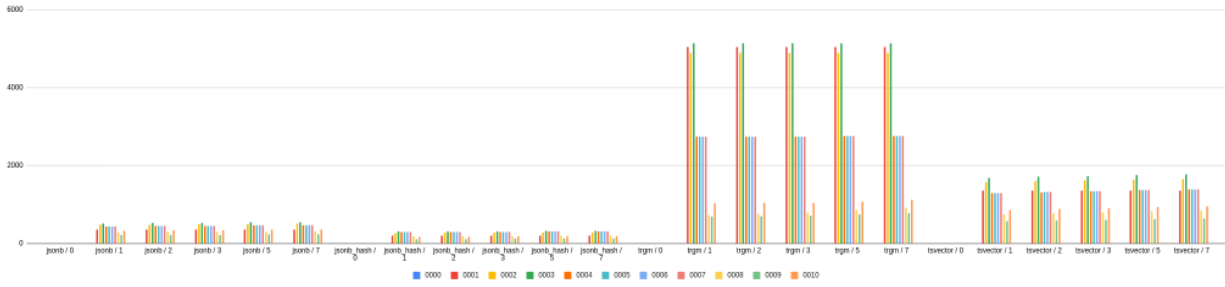


temporary space

Amount of temporary data generated by the different builds (in MB). The 0000-master never generates any temporary files (same as serial builds with 0 workers).

label	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	0	0	0	0	0	0	0	0	0	0	0
jsonb / 1	0	365	486	521	446	446	446	446	287	222	333
jsonb / 2	0	367	495	530	456	456	457	456	295	228	343
jsonb / 3	0	368	500	537	463	463	461	463	300	231	349
jsonb / 5	0	370	507	544	471	471	471	471	307	237	357
jsonb / 7	0	370	512	549	476	476	476	476	312	240	362
jsonb_hash / 0	0	0	0	0	0	0	0	0	0	0	0
jsonb_hash / 1	0	207	275	312	294	294	294	293	190	117	179
jsonb_hash / 2	0	208	280	318	300	300	300	300	195	120	184
jsonb_hash / 3	0	208	284	322	304	304	305	304	199	122	187
jsonb_hash / 5	0	209	288	327	310	310	310	310	203	125	192
jsonb_hash / 7	0	210	291	331	314	314	314	314	206	127	194
trgm / 0	0	0	0	0	0	0	0	0	0	0	0
trgm / 1	0	5042	4893	5136	2748	2748	2749	2749	729	693	1026
trgm / 2	0	5043	4894	5138	2749	2750	2751	2751	761	708	1041
trgm / 3	0	5045	4895	5139	2751	2752	2753	2753	792	722	1056
trgm / 5	0	5045	4897	5141	2756	2756	2757	2758	856	751	1086
trgm / 7	0	5046	4900	5143	2764	2765	2765	2765	922	782	1121
tsvector / 0	0	0	0	0	0	0	0	0	0	0	0
tsvector / 1	0	1358	1584	1695	1298	1294	1299	1298	751	577	855

tsvector / 2	0	1360	1604	1715	1320	1324	1323	1326	779	594	885
tsvector / 3	0	1363	1619	1736	1345	1340	1344	1344	802	611	906
tsvector / 5	0	1365	1644	1761	1370	1370	1369	1369	831	629	934
tsvector / 7	0	1367	1658	1780	1388	1389	1388	1387	851	644	951

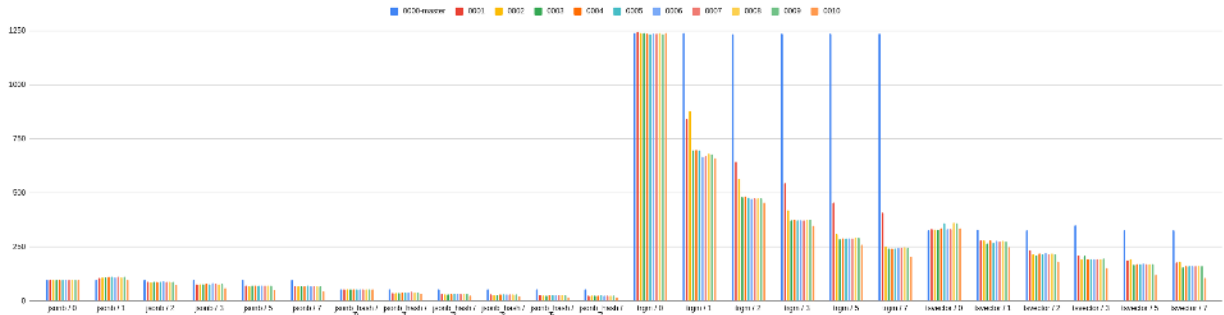


xeon / work_mem = 32MB

absolute timings

Timing in seconds for different indexes on the messages table, using a different number of parallel workers (0 - serial build). With different patches of the series applied (0000 - no patches).

label	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	97	96	98	97	97	98	96	98	96	97	97
jsonb / 1	97	105	109	109	110	111	110	111	109	110	98
jsonb / 2	96	87	87	87	89	88	91	89	89	89	73
jsonb / 3	97	78	76	77	80	78	81	79	78	78	60
jsonb / 5	98	71	69	70	71	71	72	72	71	71	50
jsonb / 7	97	69	67	67	68	70	69	68	68	68	46
jsonb_hash / 0	53	53	53	53	53	54	53	54	54	54	54
jsonb_hash / 1	53	37	37	37	40	39	40	41	39	39	33
jsonb_hash / 2	53	32	31	32	33	33	34	33	32	33	24
jsonb_hash / 3	53	29	28	28	29	30	30	31	30	30	22
jsonb_hash / 5	53	26	25	26	26	27	27	27	27	27	16
jsonb_hash / 7	53	25	24	24	25	26	26	26	25	25	15
trgm / 0	1237	1243	1238	1239	1236	1233	1236	1236	1237	1233	1239
trgm / 1	1237	839	877	695	698	694	667	668	680	678	661
trgm / 2	1234	642	565	481	482	477	471	473	475	474	454
trgm / 3	1234	544	419	373	375	374	372	372	376	377	346
trgm / 5	1237	455	309	287	289	289	289	290	290	291	258
trgm / 7	1237	408	252	242	242	242	246	246	247	246	205
tsvector / 0	327	331	330	330	335	359	332	331	360	360	334
tsvector / 1	329	281	279	266	279	269	277	275	276	275	248
tsvector / 2	327	233	216	212	220	215	221	216	218	216	181
tsvector / 3	350	211	193	206	193	193	192	194	192	195	152
tsvector / 5	328	187	192	166	170	169	172	171	169	169	119
tsvector / 7	327	178	181	155	162	162	160	161	160	161	106

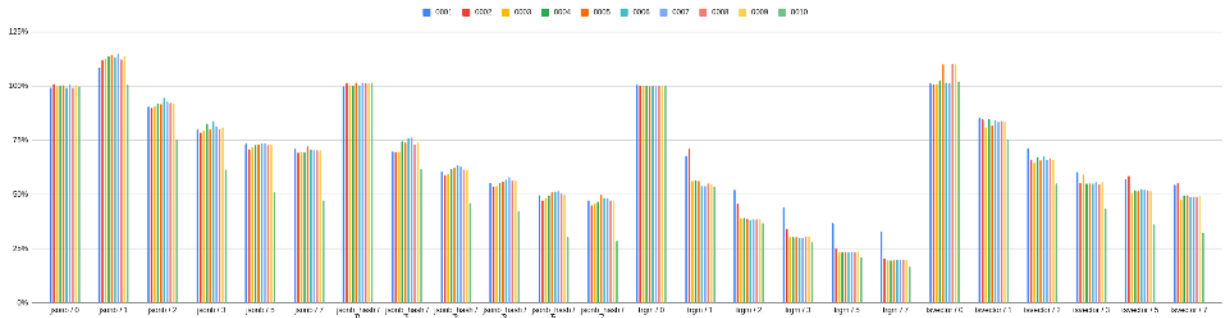


relative timings

Timing relative to 0000-master, i.e. always serial build. Ideal value (perfect parallel speedup) would be $1 / (1 + \text{workers})$, but that is not really achievable.

label	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	99%	101%	100%	100%	100%	99%	100%	99%	100%	100%
jsonb / 1	108%	112%	112%	114%	114%	113%	115%	112%	114%	101%
jsonb / 2	90%	90%	91%	92%	91%	94%	93%	92%	92%	75%
jsonb / 3	80%	79%	80%	83%	80%	84%	81%	80%	81%	61%
jsonb / 5	73%	71%	72%	73%	73%	73%	74%	73%	73%	51%
jsonb / 7	71%	69%	69%	70%	72%	71%	70%	70%	70%	47%
jsonb_hash / 0	100%	101%	100%	100%	102%	100%	101%	101%	101%	102%
jsonb_hash / 1	70%	70%	70%	75%	74%	76%	76%	73%	74%	62%
jsonb_hash / 2	61%	59%	59%	62%	62%	64%	63%	61%	61%	46%
jsonb_hash / 3	55%	54%	54%	55%	56%	57%	58%	57%	57%	42%
jsonb_hash / 5	50%	47%	48%	50%	51%	51%	52%	50%	50%	30%
jsonb_hash / 7	47%	45%	46%	47%	50%	48%	48%	47%	47%	29%
trgm / 0	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
trgm / 1	68%	71%	56%	56%	56%	54%	54%	55%	55%	53%
trgm / 2	52%	46%	39%	39%	39%	38%	38%	39%	38%	37%
trgm / 3	44%	34%	30%	30%	30%	30%	30%	30%	31%	28%
trgm / 5	37%	25%	23%	23%	23%	23%	23%	23%	24%	21%
trgm / 7	33%	20%	20%	20%	20%	20%	20%	20%	20%	17%
tsvector / 0	101%	101%	101%	102%	110%	101%	101%	110%	110%	102%
tsvector / 1	85%	85%	81%	85%	82%	84%	84%	84%	84%	75%
tsvector / 2	71%	66%	65%	67%	66%	67%	66%	66%	66%	55%
tsvector / 3	60%	55%	59%	55%	55%	55%	56%	55%	56%	43%

tsvector / 5	57%	59%	51%	52%	51%	52%	52%	52%	52%	36%
tsvector / 7	54%	55%	47%	49%	50%	49%	49%	49%	49%	32%



temporary space

Amount of temporary data generated by the different builds (in MB). The 0000-master never generates any temporary files (same as serial builds with 0 workers).

label	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010
jsonb / 0	0	0	0	0	0	0	0	0	0	0	0
jsonb / 1	0	341	474	507	432	432	432	432	286	221	333
jsonb / 2	0	344	483	517	443	442	442	443	295	227	342
jsonb / 3	0	345	488	523	448	450	448	450	300	231	349
jsonb / 5	0	347	496	532	458	458	458	458	307	236	356
jsonb / 7	0	349	501	537	463	464	463	463	311	239	361
jsonb_hash / 0	0	0	0	0	0	0	0	0	0	0	0
jsonb_hash / 1	0	195	269	306	286	286	285	286	189	116	179
jsonb_hash / 2	0	196	274	311	292	293	292	292	195	120	183
jsonb_hash / 3	0	197	278	316	297	297	297	297	198	122	187
jsonb_hash / 5	0	198	282	321	303	303	303	303	203	124	191
jsonb_hash / 7	0	198	285	324	307	307	307	307	206	126	194
trgm / 0	0	0	0	0	0	0	0	0	0	0	0
trgm / 1	0	2634	3688	3713	1174	1174	1176	1176	681	673	1006
trgm / 2	0	2635	3690	3714	1175	1175	1177	1177	689	678	1011
trgm / 3	0	2635	3690	3715	1177	1177	1179	1179	696	682	1016
trgm / 5	0	2636	3692	3717	1181	1181	1183	1183	712	691	1026
trgm / 7	0	2636	3694	3719	1190	1190	1191	1191	731	702	1040
tsvector / 0	0	0	0	0	0	0	0	0	0	0	0
tsvector / 1	0	1110	1459	1550	1141	1137	1135	1135	738	569	846

tsvector / 2	0	1121	1485	1580	1171	1168	1170	1169	769	590	878
tsvector / 3	0	1127	1503	1599	1192	1190	1193	1190	791	603	896
tsvector / 5	0	1135	1528	1629	1223	1224	1224	1222	820	625	928
tsvector / 7	0	1140	1544	1647	1245	1245	1244	1245	843	639	948

