Ruby - Feature #20902

Allow `IO::Buffer#copy` to release the GVL.

11/20/2024 07:31 AM - ioquatix (Samuel Williams)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	

Description

Related to https://bugs.ruby-lang.org/issues/20876.

Background

IO::Buffer#copy execution time is proportional to the length of the data copied. As such, large copies can take a long time (100ms+). Currently, the GVL is not released, which can stall the Ruby interpreter.

Proposal

Pull Request: https://github.com/ruby/ruby/pull/12021

If the size of the data to be copied is larger than a specific amount (heuristic), we will perform memmove using rb_nogvl.

The initial size heuristic is set to 1MiB. This won't be perfect for every system, but should be good enough to avoid ms+ stalls.

Results

I measured the difference:

GVL	Threads	Buffer Size	Total Duration	Throughput (MB/s)
Yes	1	1	0.12ms	8393.09
Yes	1	5	0.51ms	9857.7
Yes	1	10	1.12ms	8937.54
Yes	1	20	2.22ms	9015.95
Yes	2	1	0.24ms	8307.07
Yes	2	5	1.13ms	8819.58
Yes	2	10	1.49ms	13385.35
Yes	2	20	5.63ms	7110.8
Yes	4	1	0.92ms	4360.18
Yes	4	5	2.08ms	9606.58
Yes	4	10	4.51ms	8863.13
Yes	4	20	9.3ms	8601.41
Yes	8	1	1.22ms	6574.93
Yes	8	5	3.56ms	11239.27
Yes	8	10	7.31ms	10943.68
Yes	8	20	15.57ms	10274.99
Yes	16	1	1.95ms	8220.16
Yes	16	5	5.51ms	14518.05
Yes	16	10	13.77ms	11618.96
Yes	16	20	27.21ms	11759.43
Yes	32	1	3.24ms	9891.05

GVL	Threads	Buffer Size	Total Duration	Throughput (MB/s)
Yes	32	5	11.42ms	14007.41
Yes	32	10	21.64ms	14786.48
Yes	32	20	45.52ms	14060.25
No	1	1	0.13ms	7582.85
No	1	5	0.44ms	11248.55
No	1	10	1.11ms	9029.91
No	1	20	2.43ms	8228.42
No	2	1	0.18ms	11245.61
No	2	5	0.96ms	10396.76
No	2	10	1.9ms	10501.59
No	2	20	3.16ms	12656.77
No	4	1	0.69ms	5827.76
No	4	5	1.15ms	17440.54
No	4	10	2.31ms	17307.79
No	4	20	4.11ms	19483.68
No	8	1	0.67ms	11954.1
No	8	5	1.3ms	30713.68
No	8	10	2.05ms	38990.98
No	8	20	4.15ms	38552.37
No	16	1	0.96ms	16698.03
No	16	5	1.46ms	54782.47
No	16	10	2.74ms	58295.64
No	16	20	4.89ms	65482.43
No	32	1	1.82ms	17554.27
No	32	5	2.68ms	59673.59
No	32	10	3.87ms	82733.34
No	32	20	6.93ms	92297.47

In the base case, the performance is about the same, but in the best case, the throughput is significantly better: 15GiB/s vs 92GiB/s (32 threads copying 20MiB of data).

Associated revisions

Revision 3c0b09ac9e9afe5a8a536ac3c27c9202bcebc25c - 11/20/2024 08:27 AM - Samuel Williams

Allow io_buffer_memmove to release the GVL for large buffers. (#12021)

[Feature #20902]

Revision 3c0b09ac9e9afe5a8a536ac3c27c9202bcebc25c - 11/20/2024 08:27 AM - Samuel Williams

Allow io_buffer_memmove to release the GVL for large buffers. (#12021)

[Feature #20902]

Revision 3c0b09ac - 11/20/2024 08:27 AM - Samuel Williams

Allow io_buffer_memmove to release the GVL for large buffers. (#12021)

[Feature #20902]

History

#1 - 11/20/2024 07:33 AM - ioquatix (Samuel Williams)

In addition to this proposal, which is limited to IO::Buffer, maybe we should consider introducing a general rb_memmove which releases the GVL according to the same heuristic driven approach. However, I feel that is a bigger proposal and outside the scope of this feature.

#2 - 11/20/2024 09:19 AM - Anonymous

- Status changed from Open to Closed

Applied in changeset git|3c0b09ac9e9afe5a8a536ac3c27c9202bcebc25c.

Allow io_buffer_memmove to release the GVL for large buffers. (#12021)

[Feature #20902]