Performance Improvements in Maple 2021

Maple 2021 improves the performance of many routines.

map and some related commands

- For Maple 2021, the map command and some other built-in commands related to it (such as element-wise operators and zip) have been sped up significantly for common operations. In particular, there is a facility that recognizes some operations that are frequently mapped, and uses a faster implementation for these operations in cases where that is applicable. Here are some examples. We ran these on one particular machine in Maple 2020 and 2021. To be precise, we ran each call to CodeTools:-Usage in a new session, defining **L** each time, rather than in a single session as you see in this worksheet. The numbers shown after each command give the *factor of reduction in memory used*, *the speedup in CPU time*, and *the speedup in real time*.
- The following were computed for **`L := [seq(-10^6..10^6)]`**

Operation	Factor of Reduction in Memory Used	Speedup in CPU Time	Speedup in Real Time
`map(`-`, L)`	14	63	62
`map(abs, L)`	8.2	29	28
`map(has, L, 42)`	10	42	42
`map(hastype, L, posint)`	10	35	34
`map(type, L, posint)`	10	40	40
`map(`+`, L, 42)`	4.5	14	14
`map(`-`, L, 42)`	4.5	15	15
`map(`*`, L, 42)`	4.5	15	15
`map(`/`, L, 1/3)`	4.5	13	13
`map(`^`, L, 2)`	1.5	6.5	6.6

- The following were computed for `L := [seq(true, 1..10^6), false]`

Operation	Factor of Reduction in Memory Used	Speedup in CPU Time	Speedup in Real Time
`map(`not`, L)`	10	55	56
`map(`and`, L, true)`	80000 *	54	54
`map(`or`, L, true)`	10	47	46
`map(`xor`, L, true)`	15	56	59

Note: The result (*) in the preceding table occurs because Maple manages to do this with almost no memory used in the current implementation.

- The following were computed for `L := [seq([i,i+1], i = -10^6..10^6)]`

Operation	Factor of Reduction in Memory Used	Speedup in CPU Time	Speedup in Real Time
`map(has, L, 42)`	10	28	17
`map(hastype, L, posint)`	10	22	13
`map(type, L, posint)`	10	34	21
`map(op, L)`	3.3	45	26
`map2(op, 1, L)`	10	32	20
`map2(op, 2, L)`	10	35	21
`map(min, L)`	2.7	1.7	1.7
`map(max, L)`	2.7	1.7	1.7
`map(`?[]`, L, [1])`	13	66	40
`map(`?[]`, L, [2])`	13	67	41

- The following were computed for `L := [seq(i = i+1, i = -10^6..10^6)]`

Operation	Factor of Reduction in Memory Used	Speedup in CPU Time	Speedup in Real Time
`map(lhs, L)`	8.0	33	21
`map(rhs, L)`	8.0	26	17

- The following were computed for **`L := [seq({i, 42}, i = -10^6..10^6)]`**

Operation	Factor of Reduction in Memory Used	Speedup in CPU Time	Speedup in Real Time
`map(`intersect`, L, {2})`	4.0	2.0	2.1
`map(`union`, L, {2})`	1.7	1.6	1.6
`map2(member, 3, L)`	10	32	20

- The following were computed for **`L := [seq("abc", 1..10^6)]`**

Operation	Factor of Reduction in Memory Used	Speedup in CPU Time	Speedup in Real Time
`map(`?[]`, L, [1])`	13	28	29

- The differences in speedup between CPU time and real time, where present, are almost entirely caused by a reduction in garbage collection. Garbage collection is parallelized, so it takes more CPU time than real time; a reduction in garbage collection therefore benefits the CPU time more than the real time.

GMP Update

- The version of the GMP library used for large integer arithmetic by Maple has been upgraded to version 6.2.0. This adds improved support for new hardware, leading to small performance improvements for various integer operations. Micro benchmarks for the basic integer operations (addition, multiplication, gcd, etc) show speedups in the range 5-10%.

Binomial Coefficients

- The underlying algorithm to compute $\binom{a}{b}$ for integer values of a and b in the binomial command has been improved in Maple 2021. Previously a recurrence formula was used that minimized the size of intermediate calculations, but could be very slow for large values of b. Now the result is either computed more directly, or computed using the GMP implementation of **`binomial`**. The following computation is about 100 times faster in Maple 2021 than in Maple 2020.

```
> CodeTools:-Usage( binomial( 200001, 50001 ) ):
memory used=0.53MiB, alloc change=0 bytes, cpu time=5.00ms, real time=5.00ms, gc time=0ns
```

- The old binomial algorithm was highly optimized for computing, in order, sequences of binomial coefficients where the value of a is fixed. That remains about the same speed

in the new implementation.

```
> CodeTools:-Usage( seq( binomial( 100000, i ), i = 1 .. 100000) ):
```

```
memory used=2.06GiB, alloc change=0.89GiB, cpu time=18.93s, real
time=12.30s, gc time=13.00s
```

- But now, it is also fast for sequences that are out of order or with fixed values of b. The following sequences are extremely slow to compute in previous versions of Maple.

```
> a := 10^5-4; the_bs := Statistics:-Shuffle( [seq(i, i=1..a, 10^4)]
  ):
```

$$a := 99996 \tag{1}$$

```
> CodeTools:-Usage( seq( binomial( a, b),  b in the_bs) ):
```

```
memory used=398.33KiB, alloc change=0 bytes, cpu time=13.00ms,
real time=13.00ms, gc time=0ns
```

- Computing a 10000 element sequence is only about 100 times the cost of computing the first element.

```
> CodeTools:-Usage( seq( binomial( 10000+i, 5000 ), i = 1 .. 9999) )
  :
```

```
memory used=66.13MiB, alloc change=19.99MiB, cpu time=559.00ms,
real time=485.00ms, gc time=129.62ms
```

Multinomial Coefficients

- The algorithm for computing multinomial coefficients in the command combinat:-multinomial has been improved so that it is computed using the binomial command and thus is now much faster than the previous algorithm which used a formula in terms of factorials. Examples like the following are now at least three times faster than in Maple 2020.

```
> CodeTools:-Usage(combinat:-multinomial(104000, 11000, 21000,
  31000, 41000)):
```

```
memory used=79.30KiB, alloc change=0 bytes, cpu time=3.00ms, real
time=2.00ms, gc time=0ns
```

- The following example is 10 times faster in Maple 2021 than in Maple 2020.

```
> CodeTools:-Usage( combinat:-multinomial(1501500, seq(2000 .. 1000,
  -1)) ):
```

```
memory used=94.90MiB, alloc change=2.89MiB, cpu time=487.00ms,
real time=487.00ms, gc time=0ns
```

- And special cases are now identified and computed almost instantly, avoiding the large factorial and integer division computations that result from a computation directly from

the factorial formula.

```
> combinat:-multinomial(10^10+2, 10^10, 1, 1);
```

$$100000000030000000002 \quad (2)$$

The Units[Simple] package

- Operations on numeric expressions (or more precisely, complex extended numeric expressions) in the Simple Units environment, provided by the Units[Simple] package, are much faster in Maple 2021. Here is an example.

```
> with(Units):
```

```
Automatically loading the Units[Simple] subpackage
```

```
> CodeTools:-Usage(Vector(10^4,i->sin(1.0*i),datatype=float[8])):
```

```
memory used=32.57MiB, alloc change=0 bytes, cpu time=253.00ms,
real time=252.00ms, gc time=0ns
```

- This example is 15 to 20 times faster and uses 15 to 20 times less memory than in Maple 2020.

GraphTheory

- A number of commands in the **GraphTheory** package perform faster in Maple 2021 compared to previous versions of Maple. These performance improvements are noted in the What's New in Graph Theory? help page.

PolyhedralSets

- The **Project** command in the **PolyhedralSets** package now uses a faster algorithm that checks for redundant constraints. For more information, see What's New in Polyhedral Sets?