

並列ガベージコレクタ

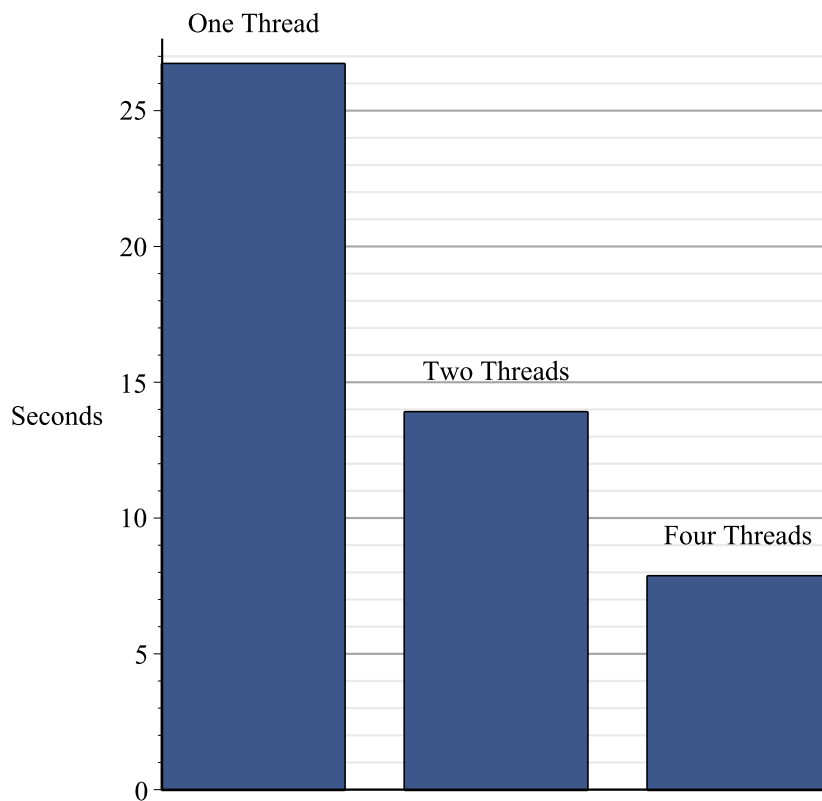
ガベージコレクタは、Maple エンジンのコアコンポーネントの 1 つです。ガベージコレクタは、評価エンジンが必要としなくなったメモリを見つけて回収します。Maple 17 では、ガベージコレクタで複数のプロセッサを使用できるようになり、ジョブをより高速に処理できるようになりました。ガベージコレクタは Maple の実行中に継続的に使用されるため、この高速化は、並列アルゴリズムを実行するユーザだけでなく、すべてのユーザに役立ちます。

- Maple 17 のガベージコレクタでは、複数コアを活用するために、複数スレッドを実行することができます。これにより、ガベージコレクタの速度を大幅に向上させることができます。ガベージコレクタは Maple の合計実行時間の一部にしか影響しませんが、それでも最大で 10% の実行時間の短縮を図ることができます。この効率化に関して、ユーザコードを変更する必要はありません。
- 並列ガベージコレクタを制御するための 2 つのカーネルオプション ([kernelopts](#)) があります。
 - [gcmmaxthreads](#) は、並列ガベージコレクタが使用する最大スレッド数を制御します。
 - [gcthreadmemorysize](#) は、Maple がガベージコレクションに対して使用するスレッド数を制御します。割り当てられたバイト数を [gcthreadmemorysize](#) に割り当てられた値で割って、スレッド数を決定します。

次の例は、複数スレッドを使用したガベージコレクタの効率化を示しています。マシンの仕様によっては、実行に数分間かかる場合があります。

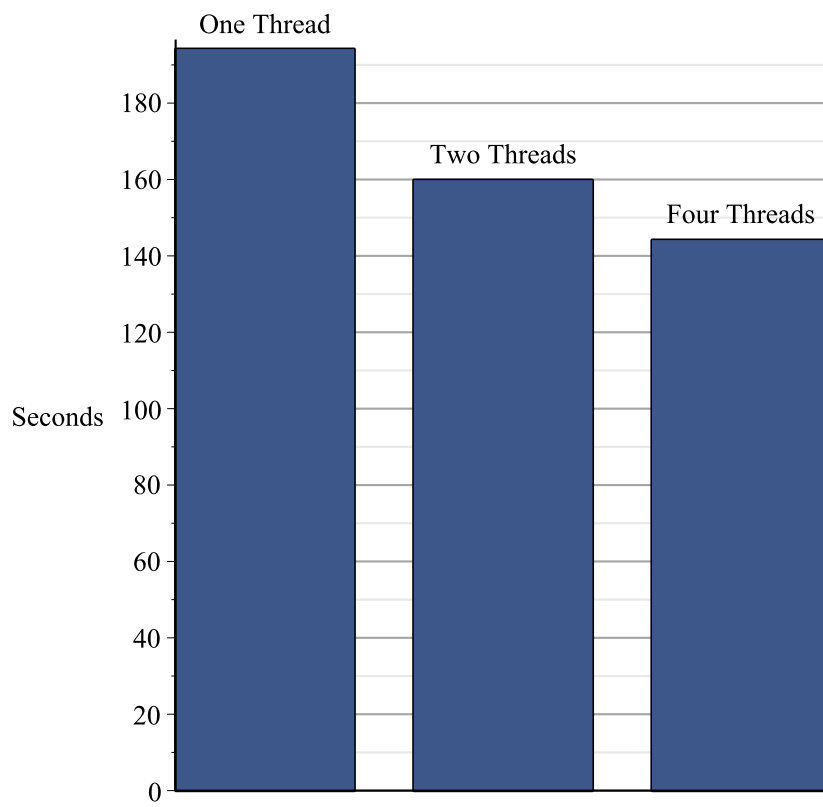
```
> restart
> if (kernelopts(wordsize) = 32) then
    upperBound := 2·10^7;
else
    upperBound := 10^8;
end if;
> data := [seq(i, i = 1 .. upperBound)]:
> p :=proc( )
    local i;
    for i to 100
    do
        gc( )
    end do
end proc;
> kernelopts(gcmmaxthreads = 1) :
    t1[1] := time[real](p( )):
> kernelopts(gcmmaxthreads = 2) :
    t1[2] := time[real](p( )):
> kernelopts(gcmmaxthreads = 4) :
    t1[4] := time[real](p( )):
```

> *Statistics:-ColumnGraph*([t1[1], t1[2], t1[4]], datasetlabels = ["One Thread", "Two Threads", "Four Threads"], labels = ["", "Seconds"], gridlines)



ガベージコレクタアルゴリズムは Maple のトータル実行時間のほんの一部しか占めません。そのため、Maple での実際の計算時間を大幅に短縮することはありません。次の例は、より現実的な計算時間の効率化を示します。

```
restartd := [seq(randpoly([x, y, z], dense, degree = 12), i = 1 .. 10000)]: f := proc( )
  local i;
  map(i → int(i, x), d)
end proc: kernelopts(gcmaxthreads = 1): t2[1] := time[real](f( )):
kernelopts(gcmaxthreads = 2): t2[2] := time[real](f( )): kernelopts(gcmaxthreads = 4): t2[4]
:= time[real](f( )):
```



参照
[kernelopts](https://kernelopts.com/)