PERFORMANCE ANALYSIS

Computer characteristics

- Computer: 11A001E424171461.
- Operative system: Windows
- Version: 10.
- Processor: Intel64 Family 6 Model 60 Stepping 3, GenuineIntel
- CPU: Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz

Software analysis

The process took 0.281000 seconds to execute

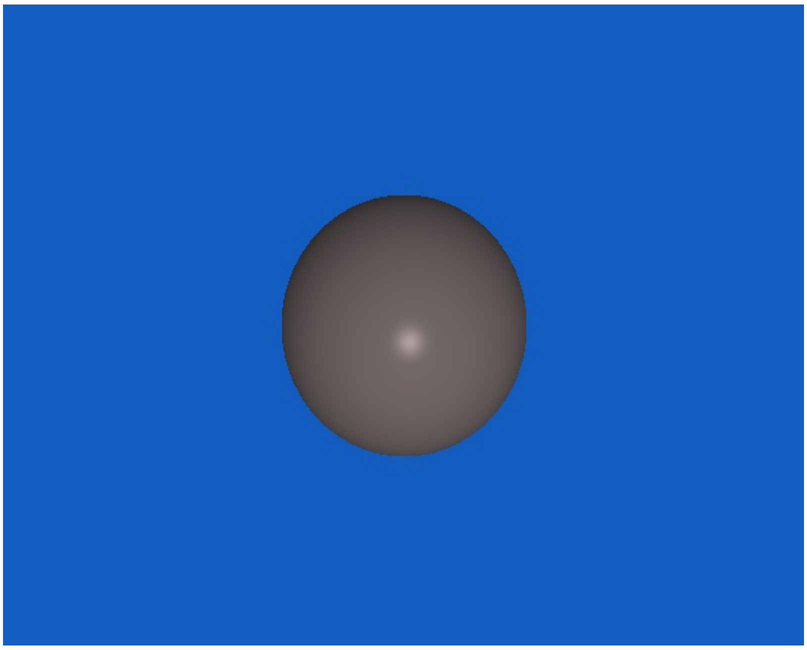


Table 1

This table gives us the following information:

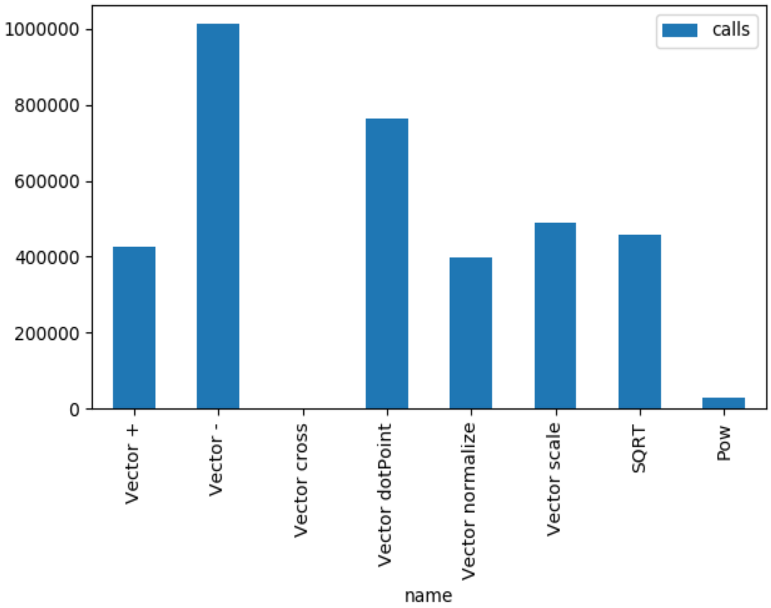
- time: the percentage of the total running time of the program used by this function.
- Accumulative seconds: a running sum of the number of seconds accounted for by this function and those listed above it.
- Self seconds: the number of seconds accounted for by this function alone. This is the major sort for this listing.
- Calls: the number of times this function was invoked, if this function is profiled, else blank.
- Self ms/call: the average number of milliseconds spent in this ms/call function per call, if this function is profiled, else blank.
- total ms/call: the average number of milliseconds spent in this function and its descendents per call, if this function is profiled, else blank.
- Name: the name of the function.

Each sample counts as 0.01 seconds.

Time	Accumulative seconds	Self seconds	calls	ms/call	ms/call	name
14.29	0.07	0.01	427783.0	0.0	0.0	Vector +
0.00	0.07	0.00	1012038.0	0.0	0.0	Vector -
0.00	0.07	0.00	2.0	0.0	0.0	Vector cross
0.00	0.07	0.00	764618.0	0.0	0.0	Vector dotPoint

Time	Accumulative seconds	Self seconds	calls	ms/call	ms/call	name
0.00	0.07	0.00	397468.0	0.0	0.0	Vector normalize
0.00	0.07	0.00	488387.0	0.0	0.0	Vector scale
0.00	0.07	0.00	457757.0	0.0	0.0	SQRT
0.00	0.07	0.00	29976.0	0.0	0.0	Pow

To get a better understanding, some table columns were graphed



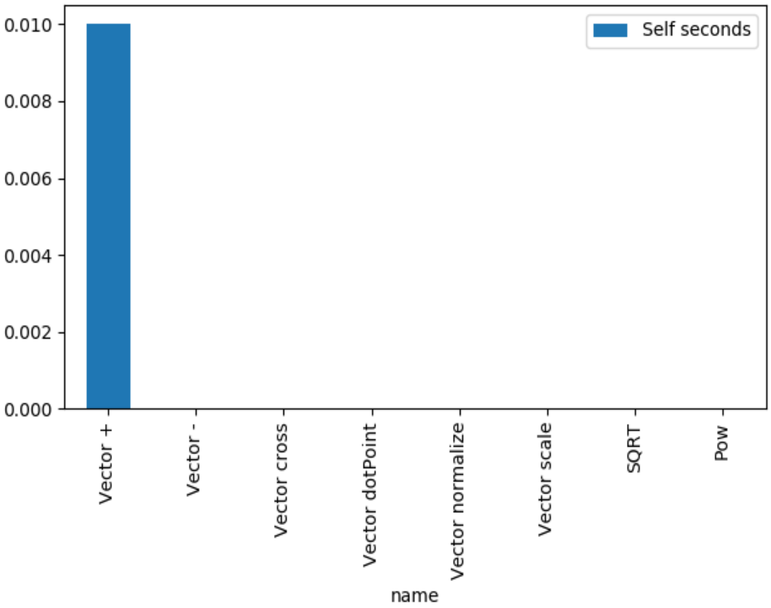


Table 2

This table describes the call tree of the program, and was sorted by the total amount of time spent in each function and its children. Each entry in this table consists of several lines. The line with the index number at the left hand margin lists the current function. The lines above it list the functions that called this function, and the lines below it list the functions this one called. This line lists:

- index: A unique number given to each element of the table. Index numbers are sorted numerically. The index number is printed next to every function name so it is easier to look up where the function is in the table.
- time: This is the percentage of the 'total' time that was spent in this function and its children. Note that due to different viewpoints, functions excluded by options, etc, these numbers will NOT add up to 100%.
- self: This is the total amount of time spent in this function.
- children: This is the total amount of time propagated into this function by its children.
- called: This is the number of times the function was called. If the function called itself recursively, the number only includes non-recursive calls, and is followed by a '+' and the number of recursive calls.

- total ms/call: the average number of milliseconds spent in this function and its descendents per call, if this function is profiled, else blank.
- total ms/call: the average number of milliseconds spent in this function and its descendents per call, if this function is profiled, else blank.
- Name: the name of the function.

For the function's parents, the fields have the following meanings:

- self This is the amount of time that was propagated directly from the function into this parent.
- children This is the amount of time that was propagated from the function's children into this parent.
- called This is the number of times this parent called the function '/' the total number of times the function was called. Recursive calls to the function are not included in the number after the '/'.
- name This is the name of the parent. The parent's index number is printed after it. If the parent is a member of a cycle, the cycle number is printed between the name and the index number.

Table 2.1

index	%time	self	children	called	name
[1]	100.0	0.00	0.07	1	RayTracer::trace(int)
		0.00	0.04	307200/337345	RayTracer::intersectionTest(Ray*, Primitive**)
		0.00	0.02	307200/307200	Ray::Ray(Vector, Vector)
		0.00	0.01	307200/307200	RayTracer::shading(Ray, Primitive*, int)
		0.01	0.00	307200/427783	Vector +
		0.00	0.00	307680/488387	Vector scale
		0.00	0.00	307200/1012038	Vector -
		0.00	0.00	307200/2017821	Vector::Vector(Vector const&)
		0.00	0.00	307200/674690	Ray::Ray(Ray const&)
		0.00	0.00	307200/307203	Color::operator*(float const&) const
		0.00	0.00	2/3066271	Vector::Vector()

Table 2.2

index	%time	self	children	called	name
		0.00	0.00	2/3066271	Vector::Vector()
[2]	100.0	0.00	0.07		_fu3___ZSt4cout
		0.00	0.07	1/1	RayTracer::trace(int)
		0.00	0.00	1/1	RayTracer::~RayTracer()

Table 2.3

index	%time	self	children	called	name
		0.00	0.00	30145/337345	RayTracer::fullScale(Vector, Vector, Vector, Color, Color, float)
		0.00	0.04	307200/337345	RayTracer::trace(int)
[3]	58.2	0.00	0.04	337345	RayTracer::intersectionTest(Ray*, Primitive**)
		0.03	0.00	337345/337345	Sphere::rayIntersection(Ray*, Primitive**)
		0.01	0.00	674690/674695	std::vector<Primitive*, std::allocator<Primitive*> >::size() const
		0.00	0.00	337345/674690	Ray::Ray(Ray const&)
		0.00	0.00	337345/337345	std::vector<Primitive*, std::allocator<Primitive*> >::operator[](unsigned int)

Table 2.4

index	%time	self	children	called	name

index	%time	self	children	called	name
		0.03	0.00	337345/337345	RayTracer::intersectionTest(Ray*, Primitive**)
[4]	43.9	0.03	0.00	337345	Sphere::rayIntersection(Ray*, Primitive**)
		0.00	0.00	30314/60459	Ray::setP()
		0.00	0.00	674690/2017821	Vector::Vector(Vector const&)
		0.00	0.00	674690/764618	Vector dotPoint
		0.00	0.00	337345/1012038	Vector -
		0.00	0.00	60289/457757	SQRT

Table 2.5

index	%time	self	children	called	name
		0.00	0.00	2/397468	RayTracer::ScreenItrInfo()
		0.00	0.00	30145/397468	Sphere::getNormal(Vector)
		0.00	0.00	60121/397468	RayTracer::fullScale(Vector, Vector, Vector, Color, Color, float)
		0.00	0.02	307200/397468	Ray::Ray(Vector, Vector)
[5]	28.6	0.00	0.02	397468	Vector normalize
		0.02	0.00	397468/397468	Vector::operator/(float const&) const
		0.00	0.00	397468/397468	Vector::lenght()

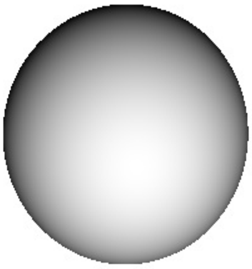
Table 2.6

index	%time	self	children	called	name
		0.02	0.00	397468/397468	Vector normalize
[6]	28.6	0.02	0.00	397468	Vector::operator/(float const&) const
		0.00	0.00	397468/2325678	Vector::Vector(float, float, float)

Hardware analysis

The process took 0.018000000000000002 seconds to execute

The image generated by the FPGA is the following one:



.