

Forum:[Hack Forums](#)

Thread: [Threading](#)

zapfen - May 8, 2013, 8:40 am

Hey Guys,

I have a short question :D

Im new in Threading and want to know some things..

1) How many Threads should be maximal used?

- I heard that it is good to use one Thread per CPU Core

2) How is Threading most efficient?

- For example a Hash Cracker: are the Passwords splitted in eg. 4 Lists and then Every Thread uses one list to crack?

- Or is there a better way?

Thanks for your Answers :)

-Zapfen

vitafortis - May 8, 2013, 8:55 am

What language are you using? Just curious.

zapfen - May 8, 2013, 9:01 am

(May 8, 2013 08:55 AM)vitafortis Wrote: [\[CITE\]](#) What language are you using? Just curious.

Im using Python :)

But im Asking in the Coding section because i think there are coders from any language :P

-Zapfen

vitafortis - May 8, 2013, 9:08 am

(May 8, 2013 09:01 AM)zapfen Wrote: [\[CITE\]](#) Im using Python :)

But im Asking in the Coding section because i think there are coders from any language :P

-Zapfen

There are, but Threading can be handled differently depending on the language.

Anyway, to answer your questions:

1) Try to use as few as possible. What I mean to say is Threading should be used after optimization to further speed up processing. Certain things must be threaded, however, such as network connections. Anyway, 3-4 is a safe number, but try not to exceed, in my opinion, 6 or so. An alternative to just threading is to Thread Pool. Thread Pooling is basically pushing threads into a queue and process them as fast as possible in order, so you don't have a ton of threads going at once.

2) It's most efficient when you have a task that must do a lot of things one time. IE, I am working on a project that captures images and processes them. I put the capturing of images on one thread, and processing them on another thread. This means that processing can take place separately so you can capture as many as possible, and save them to a data structure to process at a slower

speed.

Hope this helps.

zapfen - May 8, 2013, 9:11 am

(May 8, 2013 09:08 AM)vitafortis Wrote: [\[CITE\]](#) There are, but Threading can be handled differently depending on the language.

Ignore this part :P

Yea Okay, i know how i setup Threads in Python and how to handle them
But just in general:

If i would like to code a Hash cracker(i know a bit skiddy but dont know any other example) Then you need to speed it up so you can check much more Passwords. And this is done with Threads, but how? how are the Threads used? For every Hash an own Thread? split 5000 Passwords in 5 lists with 100 Passwords each and run every single list in one Thread?

Sry for the bad explanation :S

I hope you know what i mean :)

(May 8, 2013 09:08 AM)vitafortis Wrote: [\[CITE\]](#) Anyway, to answer your questions:

1) Try to use as few as possible. What I mean to say is Threading should be used after optimization to further speed up processing. Certain things must be threaded, however, such as network connections. Anyway, 3-4 is a safe number, but try not to exceed, in my opinion, 6 or so. An alternative to just threading is to Thread Pool. Thread Pooling is basically pushing threads into a queue and process them as fast as possible in order, so you don't have a ton of threads going at once.

2) It's most efficient when you have a task that must do a lot of things one time. IE, I am working on a project that captures images and processes them. I put the capturing of images on one thread, and processing them on another thread. This means that processing can take place separately so you can capture as many as possible, and save them to a data structure to process at a slower speed.

Hope this helps.

Sry saw this answers a bit to late :S

Thanks for this, i think i understood the principle of Threads, this really helps!

-Zapfen