From: Perlner, Ray (Fed)
To: Liu, Yi-Kai (Fed)
Subject: RE: Thermodynamic analysis of brute force Key Search, Claw finding problems.
Date: Monday, July 10, 2017 11:40:00 AM

Yep. That was the point I was trying to make. I should note that the quadratic slowdown does show up at a smaller scale: The time for the whole computation scales quadratically with the depth of the circuit required to try a single key guess. Interestingly, Grover's algorithm also requires a time that scales with the square of the depth of the circuit for trying a single key guess (keeping energy budget, key space, and gates-per-depth for a single key guess fixed.) Also worth noting: Grover's algorithm's energy budget only scales with the number of gates per depth, not circuit width per se. The way I described using thermal noise to search the key space classically, the cost goes like circuit width. I suspect, though, that you can set up the potential function for your reversible circuit in such a way that only the active gates are hot and the rest of the circuit stays cold, so I think even in this respect the thermodynamically ideal cost for "thermal search" is the same as for Grover search.

The paper you linked in your point 2 looks interesting. I've been pretty much ignoring fault tolerance, since I don't think there's any hard lower limit on the amount of noise you can have in your computation architecture, but it will be interesting to consider.

-----Original Message-----
From: Liu, Yi-Kai (Fed)
Sent: Saturday, July 08, 2017 12:58 PM
To: Moody, Dustin (Fed) <dustin.moody@nist.gov>; Perlner, Ray (Fed) <ray.perlner@nist.gov>; Jordan, Stephen P (Fed) <stephen.jordan@nist.gov>; Miller, Carl A. (Fed) <carl.miller@nist.gov>
Subject: Re: Thermodynamic analysis of brute force Key Search, Claw finding problems.

Oh, whoops, actually I take back my objection in point 1. I guess your point is that in classical exhaustive search, each processor *doesn't* have to do a sequential computation of length K/M, instead it can try K/M random guesses for the key, where each random guess is a sequential computation of length O(1), and these random guesses can be done in any order. So I guess there isn't a quadratic slowdown after all. That is interesting!

From: Liu, Yi-Kai (Fed)
Sent: Saturday, July 8, 2017 12:46:32 PM
To: Moody, Dustin (Fed); Perlner, Ray (Fed); Jordan, Stephen P (Fed); Miller, Carl A. (Fed)
Subject: Re: Thermodynamic analysis of brute force Key Search, Claw finding problems.

Hi Ray,

I thought about it some more... it sounds interesting. I was wondering about a couple of things:

1. I'm a bit skeptical about one part of your calculation. For classical exhaustive key search, you are saying that most of the computation is unpowered. You use energy to initialize the memory, and to transition to the halting state; but everything else in between is unpowered. That means that the in-between steps are essentially a random walk. In particular, for each possible key, in order to check whether that key is valid, you need to do a sequential computation of length K/M. To accomplish this, you take random steps going forwards and backwards through that sequential computation. But that means that, if you run the random walk for s steps, with high probability you will only complete sqrt(s) steps of the computation. I.e., there is a quadratic slowdown, due to the random walk. I don't think your calculation accounts for this? I think you need to set s = (K/M)^2, and so the energy cost per memory unit is K^2 / M^2 t, and the total energy cost is E = K^2 / Mt. (Rather than E = K/t.)

Alternatively, you can use power to make sure the middle steps of the computation go through deterministically. Then you don't have the quadratic slowdown due to the random walk; but now the number of powered operations is K, rather than M. So again the total energy cost is E = K^2 / Mt. (Rather than E = K/t.)

You can also do the calculation for Grover's algorithm using either of these approaches. Either way, it seems like you get E = K/t, so Grover's algorithm does have an advantage over classical exhaustive search. Or am I missing something?

2. Another interesting question would be to look at the cost of doing fault-tolerant computation. I don't know that much about the classical case, but for the quantum case, there's this paper: https://arxiv.org/abs/1301.1995

Cheers,

--Yi-Kai

From: Liu, Yi-Kai (Fed)
Sent: Wednesday, July 5, 2017 1:48 PM
To: Moody, Dustin (Fed); Perlner, Ray (Fed); Jordan, Stephen P (Fed); Miller, Carl A. (Fed)
Subject: Re: Thermodynamic analysis of brute force Key Search, Claw finding problems.

Yes, that looks interesting! Let me get back to you later this week... (b) (6)

Cheers,

--Yi-Kai

From: Moody, Dustin (Fed)
Sent: Wednesday, July 5, 2017 11:47 AM
To: Perlner, Ray (Fed); Liu, Yi-Kai (Fed); Jordan, Stephen P (Fed); Miller, Carl A. (Fed)
Subject: RE: Thermodynamic analysis of brute force Key Search, Claw finding problems.

Yi-Kai, Stephen,
Have either of you been able to look at what Ray wrote? Curious as to your thoughts….

Dustin

From: Perlner, Ray (Fed)
Sent: Friday, June 30, 2017 11:22 AM
To: Liu, Yi-Kai (Fed) <yi-kai.liu@nist.gov>; Jordan, Stephen P (Fed) <stephen.jordan@nist.gov>; Miller, Carl A. (Fed) <carl.miller@nist.gov>; Moody, Dustin (Fed) <dustin.moody@nist.gov>
Subject: Thermodynamic analysis of brute force Key Search, Claw finding problems.

It's a well-known fact (https://cr.yp.to/hash/collisioncost-20090517.pdf) that if you're just counting gates, the complexity of the quantum collision finding algorithm from Brassard Hoyer and Tapp is no better than the best classical algorithm (Van Oorschot Weiner.) I believe the same argument indicates that Tani's claw finding algorithm doesn't really buy you anything in terms of circuit size/depth.

However, given that quantum memories can be just as energy efficient as classical memories at least in theory (see e.g. https://arxiv.org/abs/0708.1879) I was wondering whether maybe these algorithms could buy you something thermodynamically. It appears they can't:

I compared a classical reversible implementation of Van Oorschot-Weiner to a reversible version of quantum claw finding/ collision search: Recall (see e.g. https://www.math.ucsd.edu/~sbuss/CourseWeb/Math268_2013W/Bennett_Reversibiity.pdf) that the energy per operation in a reversible computation goes like the circuit depth, divided by the physical time required to compute. To find collisions/claws in a range of size N, using a memory of size M, in time t, the quantum algorithm requires sqrt(N/M) gates in series (excluding memory lookups, which we're assuming have lower power requirements than ordinary gates). The energy cost per gate is therefore sqrt(N/M)/t, resulting in a total energy cost of E = sqrt(N/M)/t * sqrt(N/M) = N/(Mt). The classical algorithm requires sqrt(N) gates in total, parallelized M ways. The circuit depth is then Sqrt(N)/M resulting in an energy cost of Sqrt(N)/ (Mt) per gate, and a total energy cost of E = Sqrt(N)/ (Mt) * Sqrt(N) = N/(Mt). Needless to say this is exactly the same.

What’s even more surprising, though, is that Grover’s algorithm has the same thermodynamic requirements as an idealized classical algorithm for key search. The classical algorithm is mostly unpowered. In order to randomly sample keys, we can use thermal noise to do a random walk on the internal state of a reversible circuit that selects a key and reaches a dead end if the key is incorrect. If the key is correct, on the other hand, the circuit goes through a thermodynamically IRreversible transition to a halt state. The only parts of this algorithm that need to dissipate power are the final transition to the halt state (which requires a negligible amount of power,) and the initial construction of the circuit, which requires energy proportional to the memory times the temperature of the thermal noise used to do the random walk. In order to search a key space of size K in time t using M memory units, the energy cost per memory unit is K/Mt, and the total energy cost is E = K/Mt*M = K/t. Compare to a reversible/parallel version of Grover’s algorithm. The circuit depth is sqrt(K/M) resulting in a per gate energy cost of sqrt(K/M)/t and a total energy cost of E = sqrt(K/M)/t* sqrt(K/M)* M = K/t.

I think the above is likely a somewhat deep result, and it give a good reason to be skeptical of the utility of quantum claw finding at the very least. There is a sensible argument that Grover’s algorithm might turn out to be useful after all, since if you try to duplicate the performance of Grover’s algorithm at fixed power, but for arbitrary key size, using the classical algorithm, you either require an amount of memory that grows linearly with the time, or you need to run the system at an energy scale that grows linearly with time. (Niether of these is required for Grover.) Arbitrarily high memory requirements or temperatures are notably inconvenient. That said, my calculations indicate that you could get away with using terrestrial scale resources at room temperature to duplicate Grover’s algorithm for about a year.

Also worth noting, I neglected small factors (like the circuit size and depth of individual block cipher or hash function queries, and just set them to 1. I did work it out, not suppressing these factors, and both pairs of algorithms still came out the same, though.

Anyway, I think this is a very interesting result. I would be interested to know if any of you want to collaborate on it.

Cheers,
Ray