<Intro Par>

The area of genetic sequencing has been enjoying a period of exponential growth for several decades now. Breakthroughs in chemistry, as well as significant improvements in sequencing equipment, have led to a rapid decline in the cost of sequencing an organism's genetic data. At the same time, the increasing power of computing machinery and the decreasing cost of storage have encouraged the ever-greater accumulation of data in a large variety of areas. These two pressures have made, and will continue to make, the sequencing of DNA from large numbers of organisms more accessible. Already, there are projects in various stages of completion which aim to sequence hundreds, or even thousands, of genomes from a single species or clade. While these developments are beneficial to the fields of biology and genomics, they do present new challenges and highlight shortcomings in the existing models used to represent genetic data. The current standard in genomics is to use the genome of a single organism as a reference for its species. The increasing prevalence of multiple complete genomes per species leads to a desire for more than one reference sequence per species. Older models in computational genomics do not have the power required to adequately handle multiple reference genomes. It is now seen as desirable to create a pan-genome, a single representation of all available gene sequences from a species which can be viewed as a single entity. The SplitMEM algorithm is designed to take multiple genomic lines and convert them to a compressed de Bruijn graph pan-genome representation, which will enable the isolation of common features in the genomes so that characteristics of the entire species or clade can be identified while gene sequences specific to an individual organism can be de-emphasized.

In most modern networks their is a stable, generally deterministic, data structure in place to handle requests. Unfortunately by definition these deterministic systems are very predictable and because of their predictability they can be abused. Users who understand the nature of the network can overload the system with a series of time consuming requests which eventually bog down the network. These denial-of-service, D.O.S., attacks can occur for a variety of reasons ranging from breaching security to gaining an advantage in competitive gaming. In general D.O.S. attacks rely on knowing how the network will respond to a every unique request. In theory if they could not predict a networks reaction they could not D.O.S. the network. Knowing this the problem now becomes creating a data structure that has unpredictable timings for any given input, while still performing at a near optimal speed. This new data structure must be able to perform the same operations as the original system, while also non-deterministically, i.e. randomly, altering itself. This new system should be able to resist all attackers, even attackers who know the algorithm and the previous i/o values used.

< Closing Par>

Tao's part (quantum computation and factoring)

Compared to a classic computer with a random number generator, a quantum computer is like a computer with a random number generator integrated into its basic computing circuits. This new aspect enables the computer to perform tasks that involve specific uncertainty in every step of computation more easily, though there's no such a physical computing system existing now. On the other hand, factoring big integers is a notoriously hard problem, especially for a deterministic computer, as the only clue we have about possible factors is the changing probability of a number being a factor when the computation goes on.

In terms of factoring, the best method we have now is based on probability of a number being a factor of a number. Therefore factoring lends itself more to quantum computation than to classic deterministic computation. In order to design a coherent computing system, we need to put some constraints on the quantum physics, essentially the same way we give rules to transistors to switch status when designing a classic computing system.

There are some defining differences involved in quantum computation which need novel design of the computing system.

Referring to quantum physics, we know that a particle can be in different positions at the same time with different probabilities. However, the total probabilities of being in all positions sum up to 1. In such a light, a quantum state representing some possible outputs should have the sum of possibilities of being in all positions equal to 1. Thus only a unitary transformation is allowed in quantum computation. This guarantees total probabilities equal to 1.

As the quantum physics laws are completely reversible, despite of the arrow of time. Therefore deterministic computation can be performed on a quantum computer only if it is reversible. Thus storing intermediate results can enable us to go around the difficulty.

A quantum gate is like a logic circuit in a classic computer which produces output when reading input. When the task in specified, the algorithm can be integrated into a quantum gate. For instance, when we have a calculation as $x^a = b$ mod n, we build n and x into the gate and use a as the input.