New multiplication algorithms

David Harvey

University of New South Wales

ANTS XI, Gyeongju, Korea

(joint work with Joris van der Hoeven and Grégoire Lecerf)

Integer multiplication

Let $\mathrm{I}(n)$ = bit complexity of multiplying n-bit integers.

Classical multiplication: $\mathrm{I}(n) = O(n^2)$.

Schönhage–Strassen (1971): $\mathrm{I}(n) = O(n \log n \log \log n)$.

Fürer (2007): $\mathrm{I}(n) = O(n \log n \, K^{\log^* n})$ for some unspecified $K > 1$.

Here $\log^*$ is the iterated logarithm:

$$\log^*\left(e^{e^{e^{e^{e^{e^{e}}}}}}\right) = 7.$$

Integer multiplication

Our main results (see "Even faster integer multiplication" on arXiv):

- A new algorithm achieving

$$\mathsf{I}(n) = O(n \log n \, 8^{\log^* n}).$$

- If there are enough Mersenne primes, we can get

$$\mathsf{I}(n) = O(n \log n \, 4^{\log^* n}).$$

- Fürer's method can be optimised to achieve

$$\mathsf{I}(n) = O(n \log n \, 16^{\log^* n}),$$

  but we don't know how to do better than 16.

Polynomial multiplication

We also give improved bounds for polynomial multiplication in $\mathbf{F}_p[X]$.

In an algebraic model, we get

$$\mathsf{M}_{\mathbf{F}_p}(n) = O(n \log n \, 8^{\log^* n}).$$

(See "Faster polynomial multiplication over finite fields" on arXiv.)

No Fürer-type bounds were previously known for this problem.

The best previous result was $O(n \log n \log \log n)$.

Implementation and performance

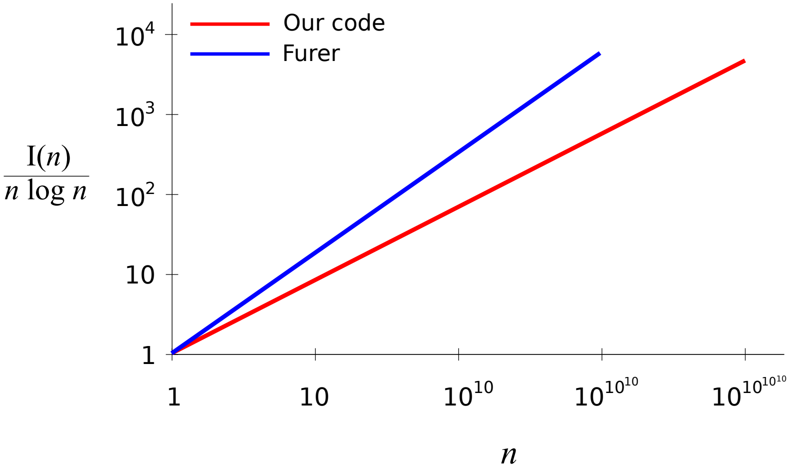
We implemented the new integer multiplication algorithm in C.

Assembly for critical inner loops.

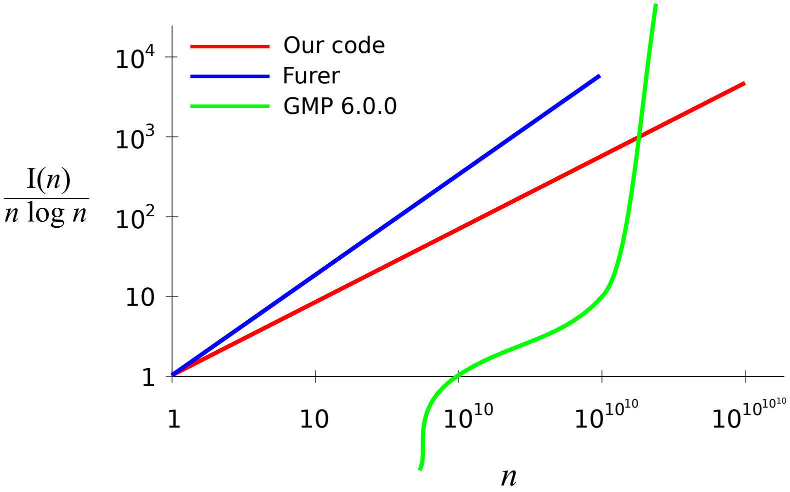
Test system:

- Customised Linux cluster
- $10^{10^{10^{10000000000}}}$ compute nodes (16 cores, 2.6 GHz, 64 GB RAM)
- Two login nodes
- Modified IP stack, permits $10^{10^{10000000000}}$-digit IP addresses

Implementation and performance



Implementation and performance



Implementation and performance