Report on the integer-factorization project at Arizona Winter School 2006

Daniel J. Bernstein

djb@cr.yp.to

Six students were officially assigned to my integer-factorization project at AWS 2006:

- Matthew Darnall (Wisconsin).
- David Freeman (Berkeley).
- Giulio Genovese (Dartmouth).
- Sonal Jain (Harvard).
- Borislaw Mezhericher (Columbia).
- Andrew Shallue (Wisconsin).

Several students started working on the project in the weeks leading up to AWS 2006. Freeman reported speeds for his ECM software (stage 1). Genovese reported speeds for his remainder-tree software. Mezhericher reported speeds for his own ECM software (stage 1).

AWS 2006 included four evening sessions devoted to collaborative work on the student projects. Each session was nominally three hours. We actually spent about twenty hours together in the evening sessions, not to mention time spent by the students outside the evening sessions.

The students gave a series of talks at the end of AWS 2006. Mezhericher and Genovese gave a 20-minute tag-team talk on trial division versus remainder trees. Freeman gave a 10-minute talk on parameter selection for ECM. Darnall gave a 10-minute talk comparing speeds of rho and ECM. Jain gave a 10-minute talk on the asymptotics of early aborts. Shallue gave a 10-minute talk on computations analyzing concrete early-abort parameters.

Here are some representative numbers from the talks. The time to find the 2^{12}-smooth part of a uniform random 256-bit integer was 1 million Athlon cycles with trial division and (in batch) 0.045 million Athlon cycles with a remainder tree. The time to find the 2^{20}-smooth part of a uniform random 256-bit integer was 52 million Athlon cycles with trial division and 0.49 million Athlon cycles with a remainder tree.

For ECM with a repeated-addition stage 2, the power-of-2 parameter choices with the best price-performance ratio were stage-1 bound 64 (multiplier $E = \operatorname{lcm}\{1, 2, ..., 64\}$) and stage-2 bound 256 (extra multipliers $65E$, $67E$, $\ldots$, $255E$) with 18% success per curve for 20-bit primes, and stage-1 bound 1024 and stage-2 bound 4096 with 3.5% success per curve for 40-bit primes. Changing 4096 to 1024, 2048, 4096, 8192, 16384 changed the success chance from 3.5% to 1.1%, 2.1%, 3.5%, 4.2%, 5.0%. The head-to-head rho-versus-ECM comparison showed rho twice as fast at finding 16-bit primes but ECM several times faster at finding 40-bit primes.

The early-abort computations used a deceptively short algorithm of Kalai to generate independent uniform random 64-bit integers represented as products of primes; this algorithm is much faster than a naive generate-and-factor. (An earlier algorithm of Bach is faster than Kalai's algorithm but not nearly as short.) The computations then identified optimal power-of-2 parameters for a single early abort. For example, a uniform random 64-bit integer has probability about $2^{-8.1}$ of being 2^{15}-smooth, probability about $2^{-3.5}$ of having 2^7-unfactored part below 2^{44}, and probability about $2^{-9.8}$ of satisfying both conditions. If detecting primes $< 2^7$ is 20 times faster per input than detecting primes $< 2^{15}$ then a unified $(2^7, 2^{44}, 2^{15})$ method, detecting about 1/3 of the 2^{15}-smooth inputs, is 7 times faster per input, i.e., more than twice as fast per output.