

Re: sparse polynomial arithmetic

Source: <http://sci.tech-archive.net/Archive/sci.math.symbolic/2008-04/msg00055.html>

- *From:* parisse@xxxxxxxxxxxxxxxx
 - *Date:* Mon, 07 Apr 2008 17:29:46 +0200
-

Now THAT is interesting. I thought you just used one big hash table.
How well does your parallel code perform ?

It is not tuned at all since I do not regularly run tests on multi-processors PC. It does launch n threads (n being the number of processors, the parent thread waits until they all finish, then it relaunch threads until all possible degrees for the main variable are computed. I should write code to synchronize them with mutexes. For dense problems, it is maybe 20% faster (as feeled by the user, I have not implemented timers that show different thread usage). I have updated giac linux binaries, implementing chinese remaindering for results with coeffs less than around 155 bits. I get $(1+x+y+z+t)^{30}$ times itself +1 in around 550 second on my athlon xp 2500+ (computation mod 2^{64} is 105s, 3 other primes required, 150s each).

You really need to buy a 2.4 GHz Core2 Quad Q6600. They're cheap and the price/performance is unbeatable. It would get you 64-bit and four cores. I would buy one if the wife would ok a 5th computer in a one bedroom apartment :)

Maybe, I'll do when I'll change my laptop, if there are multi-processor laptops available. It would be interesting to see what kind of enhancements low level arithmetic on 64 bits provide, e.g. if there is a 128 bit integer type, the test above would only use one prime. But changing my PC is currently low priority, working efficiently on multivariate polynomials with smaller coefficients is enough for a general purpose CAS. And maybe someone will do tests for me on a multiprocessor machine:—)