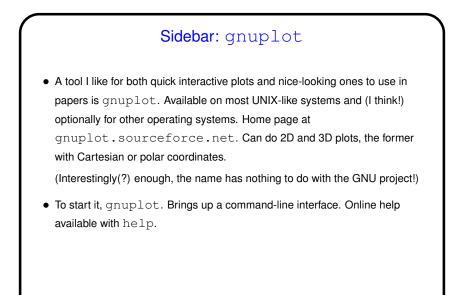
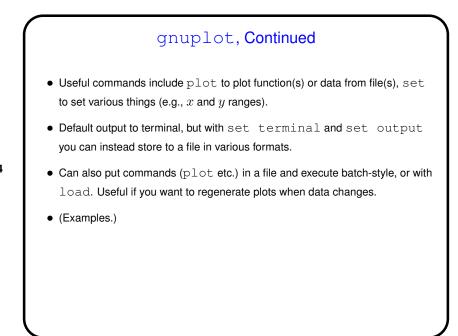


Minute Essay From Last Lecture
(How to decide whether a RNG is "good" ...)
Some people mentioned just trying it in the intended application and observing whether the results seem good. Might actually be the best way, since a RNG that's "good" for one application might be less so for another.
Others mentioned checking whether the sequence is evenly distributed, or has no repeats.
In principle, what you want is something that seems unpredictable — but keeping in mind that for at least some applications you do want results to be reproducible.



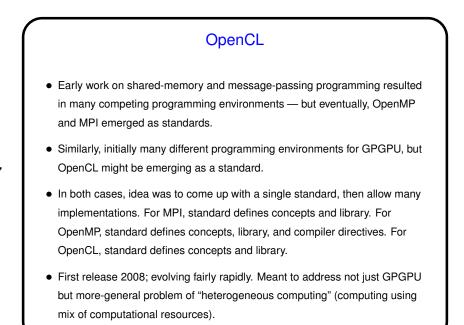


Homework 2 — Implementing LCG

- Implementing a 48-bit LCG function is doable in both C (with int 64_t and Java Long). Note, however, that the multiplication required to generate the next element can overflow which is no problem since we only want the value mod 2^{48} , *but* consider what happens if the overflow produces a negative result. Hence my suggestion to compute this with bitwise "and" (&) rather than with %.
- Implementing the described leapfrog scheme is trickier. I decided to try it in Scala first and discovered that while the modified constants a' and b' only need to be 48 bits, computing them correctly — well, my current approach is to use BigInt to do the computation and then convert the result back to a Long. This should work in Java too (with suitable changes of names), and in C... There's a library called GMP that provides support for arbitrary-precision arithmetic that looks promising. "Stay tuned"?

GPGPU
Recall from overview/introduction that the SIMD (Single Instruction, Multiple Data) model was popular in the relatively early days of parallel programming, fell of favor, and is now making a comeback as "GPGPU" (General-Purpose computing on Graphics Processing Units).
Typically SIMD is a good fit for GPU hardware — but it's worth noting that they usually(?) have their own memory, not shared with "host" CPU, which makes programming more complicated and has implications for performance.

Slide 6



What's an OpenCL Program Like?
Source code in C/C++, with calls to OpenCL functions.
Typically includes source to be compiled at runtime for whatever device is to be used. "Device"? yes, many new terms/concepts ... (And in context here it means something not exactly like what it has come to mean in popular usage!)

