GAP

Groups, Algorithms, Programming

OpenDreamKit Kickoff

GAP

- Groups, Algorithms, Programming
- Free software platform for computation in abstract algebra and discrete mathematics
    - research mathematics
    - also teaching, physics, chemistry, CS, ….
    - 1986—, Aachen, St Andrews, Braunschweig, Fort Collins, Kaiserslautern,…
        - Neubuser, Schonert
- Over 2000 citations

GAP Architecture

- Kernel — C, 160K Lines
  - read-eval-print UI, memory management, OS interface, interpreter, some performance critical routines
    - all horribly mixed up with one another
  - GAP language is imperative, dynamically typed, features multi-methods and method selection based on dynamic properties of objects
- Library — GAP, 400K lines
  - almost all the real mathematics is here
  - kernel/library interface isn’t as clean as we’d like
- Data Libraries —
  - each manages it’s own data using it’s own GAP programs, similar interfaces but no common architecture
  - some can access remote data

```
gap> AvgOrder :=
>  g->Sum(ConjugacyClasses(g),
>   c-> Size(c)*Order(Representative(c)))/
>    Size(g);
function( g ) ... end
gap> AvgOrder(MathieuGroup(11));
53131/7920
gap> ForAny(AllSmallGroups([2..100]),
> g->IsInt(AvgOrder(g)));
false
```

GAP Packages

- Much of the mathematical functionality of GAP is found in its 118 contributed packages.
    - simple addons; major pieces of software; links to external software;….
- Totalling over 1M lines of GAP code, plus many C programs and libraries

HPC-GAP

- Adapted GAP kernel and run-time for parallel computation
  - **Problem:** GAP library architecture depends heavily on updating shared data to avoid repeating work, and a call to one part of the library may lead to calls to very many other parts
    - Exactly what you don’t want with threads!
- novel architecture protects legacy code from inadvertent interference between threads
- Demonstrated performance gains and safety but only parts of the library and very little user code could be adapted within the project