Making your code go faster

Chris J. Pickard
Theory of Condensed Matter
Cavendish Laboratory
Cambridge University

UNIVERSITY OF CAMBRIDGE

The problem

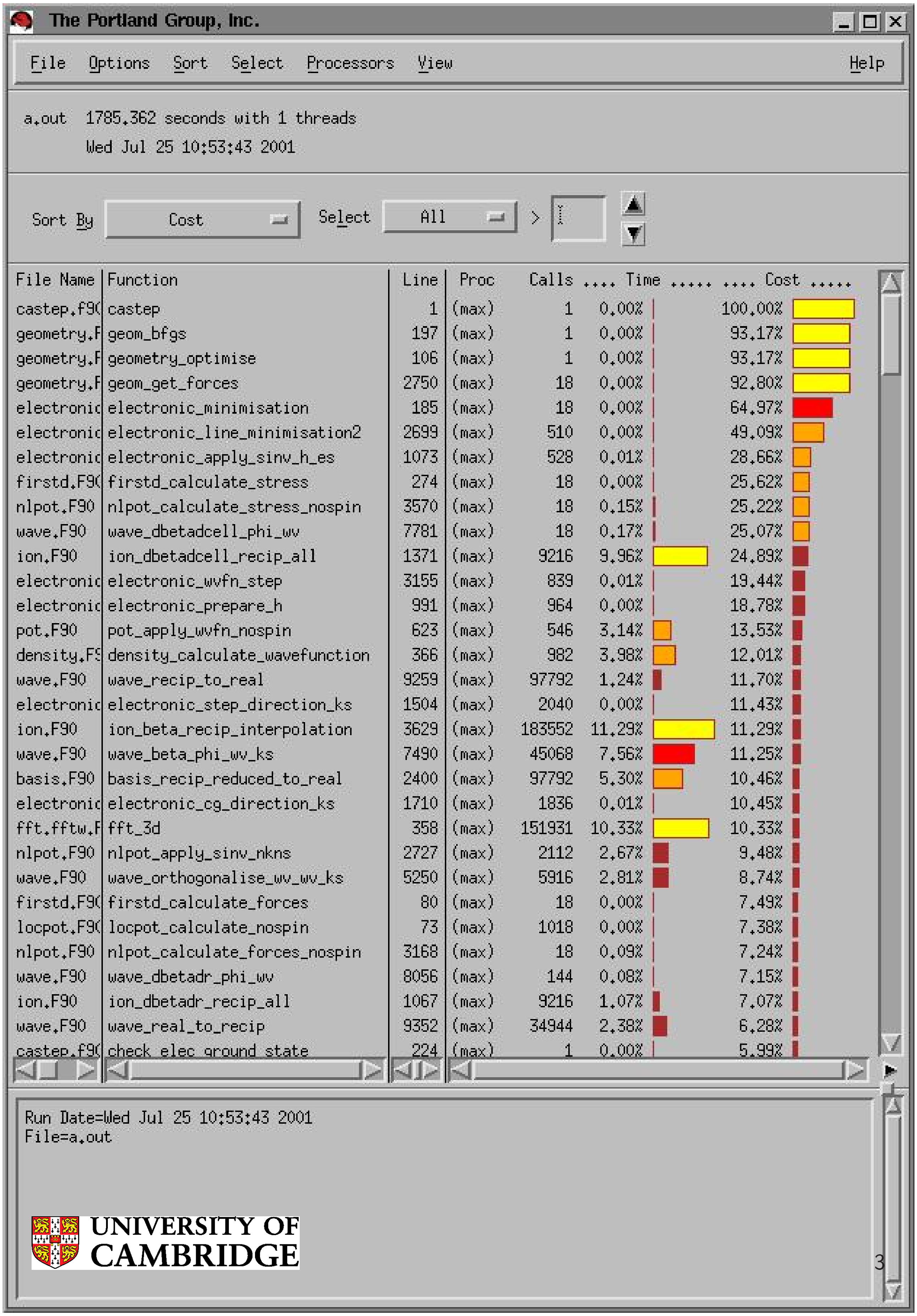
- Your code runs like a dog (newtep was about 1000 times slower than castepexe before optimisation!!!)

Seeing the problem — profiling

- On Alpha
  - compile with -pg
  - execute
  - run gprof

- On SGI (hpcf2)
  - run pixie on your executable (produce castep.pixie)
  - run castep.pixie
  - run cvperf castep.Counts

- On linux (with pgf90)
  - compile with -Mprof=func/lines
  - execute
  - run pgprof



The Portland Group, Inc.

File Options Sort Select Processors View Help

a.out 1785.362 seconds with 1 threads
Wed Jul 25 10:53:43 2001

Sort By: Cost | Select: All >

File Name	Function	Line	Proc	Calls	Time	Cost
castep.f9(	castep	1	(max)	1	0.00%	100.00%
geometry.F	geom_bfgs	197	(max)	1	0.00%	93.17%
geometry.F	geometry_optimise	106	(max)	1	0.00%	93.17%
geometry.F	geom_get_forces	2750	(max)	18	0.00%	92.80%
electronic	electronic_minimisation	185	(max)	18	0.00%	64.97%
electronic	electronic_line_minimisation2	2699	(max)	510	0.00%	49.09%
electronic	electronic_apply_sinv_h_es	1073	(max)	528	0.01%	28.66%
firstd.F9(	firstd_calculate_stress	274	(max)	18	0.00%	25.62%
nlpot.F90	nlpot_calculate_stress_nospin	3570	(max)	18	0.15%	25.22%
wave.F90	wave_dbetadcell_phi_wv	7781	(max)	18	0.17%	25.07%
ion.F90	ion_dbetadcell_recip_all	1371	(max)	9216	9.96%	24.89%
electronic	electronic_wvfn_step	3155	(max)	839	0.01%	19.44%
electronic	electronic_prepare_h	991	(max)	964	0.00%	18.78%
pot.F90	pot_apply_wvfn_nospin	623	(max)	546	3.14%	13.53%
density.F9	density_calculate_wavefunction	366	(max)	982	3.98%	12.01%
wave.F90	wave_recip_to_real	9259	(max)	97792	1.24%	11.70%
electronic	electronic_step_direction_ks	1504	(max)	2040	0.00%	11.43%
ion.F90	ion_beta_recip_interpolation	3629	(max)	183552	11.29%	11.29%
wave.F90	wave_beta_phi_wv_ks	7490	(max)	45068	7.56%	11.25%
basis.F90	basis_recip_reduced_to_real	2400	(max)	97792	5.30%	10.46%
electronic	electronic_cg_direction_ks	1710	(max)	1836	0.01%	10.45%
fft.fftw.F	fft_3d	358	(max)	151931	10.33%	10.33%
nlpot.F90	nlpot_apply_sinv_nkns	2727	(max)	2112	2.67%	9.48%
wave.F90	wave_orthogonalise_wv_wv_ks	5250	(max)	5916	2.81%	8.74%
firstd.F9(	firstd_calculate_forces	80	(max)	18	0.00%	7.49%
locpot.F9(	locpot_calculate_nospin	73	(max)	1018	0.00%	7.38%
nlpot.F90	nlpot_calculate_forces_nospin	3168	(max)	18	0.09%	7.24%
wave.F90	wave_dbetadr_phi_wv	8056	(max)	144	0.08%	7.15%
ion.F90	ion_dbetadr_recip_all	1067	(max)	9216	1.07%	7.07%
wave.F90	wave_real_to_recip	9352	(max)	34944	2.38%	6.28%
castep.f9(	check_elec_ground_state	224	(max)	1	0.00%	5.99%

Run Date=Wed Jul 25 10:53:43 2001
File=a.out

UNIVERSITY OF CAMBRIDGE

Some solutions

- Check array ordering
  - run over fastest index first
- Use vendor libraries
  - FFT (or FFTW)
  - blas/lapack (esp. zgemm)
- Do everything less often
  - can expensive objects be stored?