Scaling GalaxSee to Peta-scale Computational Resources with MPI and OpenMP

EARLHAM COLLEGE

Andrew Fitz Gibbon

Resource Architecture

Historically

- Shared Memory parallelism

Now

- Distributed Memory parallelism

Now⟶Future

- Hybrid of both

Moving from Tera-scale to Peta-scale to Exa-scale

How do we do it?

Code re-working

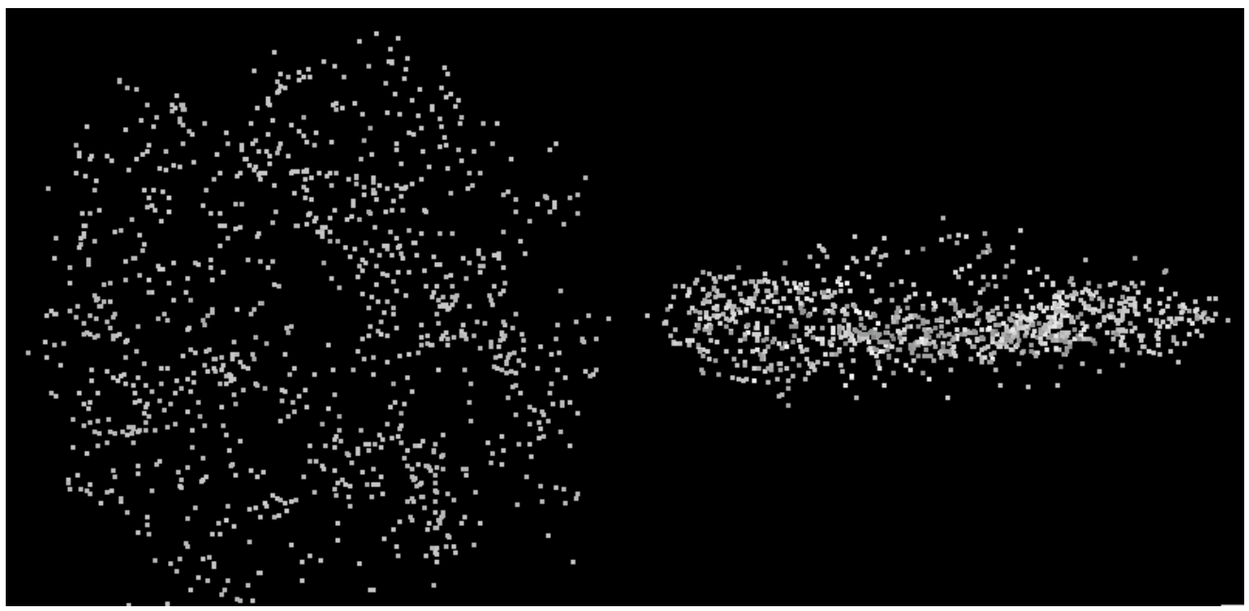
- Identify sections that are amenable to Distributed Parallelism
- Identify sections that are amenable to Shared Parallelism
- Combine the two

Simple Example: GalaxSee

- Already written for Distributed systems with MPI
- Easily profiled for analysis
- Straightforward to extend

What is GalaxSee

- Simple N-Body simulation
- Mimics the formation of a galaxy
- Developed as a tool for teaching communication intensive applications



Step 1: MPI

- Simple, uniform tiling
  - Points equally distributed to each process
- Communication after each time step
  - Processes update data for next step

Profiling

- Run on small SMP machine:
  - 4 cores per node
  - 2 processes, 2048 points, 1000 time steps
  - ~140 seconds per run

gprof data for purely-MPI runs

% time	cumulative seconds	self seconds	calls	self s/call	total s/call	name
57.43	728.99	728.99	763	0.96	0.96	derivs(int, double, double*, double*)
42.53	1268.81	539.82	508	1.06	1.06	derivs_client()
0.05	1269.46	0.65	756	0.00	0.96	diffeq::updateEuler(double, void (*)(int, double, double*, double*))
0.00	1269.50	0.04	1560576	0.00	0.00	modeldata::comp_s_rad(double, double)
0.00	1269.53	0.03	21	0.00	0.00	cart3d::init(int)
0.00	1269.55	0.02	7	0.00	0.00	diffeq::init(int)
0.00	1269.57	0.02	6	0.00	1.12	modeldata::new_galaxy()
0.00	1269.59	0.02				modeldata::calc_depth(double, double, double)
0.00	1269.59	0.00	769	0.00	0.00	mapPoints(int, int, double*, double*, cart3d*, cart3d*, cart3d*, cart3d*)
0.00	1269.59	0.00	756	0.00	0.96	run_step()
0.00	1269.59	0.00	20	0.00	0.00	MPI::Is_initialized()
0.00	1269.59	0.00	20	0.00	0.00	MPI::Comm::~Comm()
0.00	1269.59	0.00	20	0.00	0.00	MPI::Comm_Null::~Comm_Null()
0.00	1269.59	0.00	11	0.00	0.00	global constructors keyed to _Z13derivs_clientv
0.00	1269.59	0.00	11	0.00	0.00	global constructors keyed to _Z6derivsidPdS_
0.00	1269.59	0.00	11	0.00	0.00	global constructors keyed to g_dynamic

Step 2/3: Inserting OpenMP

- Identify sections of non-MPI code taking time
  - GalaxSee: “for” loops calculating accel/pos
- Thread with OpenMP
  - Simple “parallel for”

gprof data for hybrid-MPI-OpenMP runs

% time	cumulative seconds	self seconds	calls	self s/call	total s/call	name
59.51	458.94	458.94	644	0.71	0.71	global constructors keyed to _Z13derivs_clientv
40.38	770.35	311.41	517	0.60	0.60	global constructors keyed to _Z6derivsidPdS_
0.06	770.80	0.45	504	0.00	0.60	diffeq::updateEuler(double, void (*)(int, double, double*, double*))
0.05	771.15	0.35	508	0.00	0.60	derivs(int, double, double*, double*)
0.02	771.27	0.12	635	0.00	0.71	derivs_client()
0.00	771.29	0.02				modeldata::calc_depth(double, double, double)
0.00	771.30	0.01	12	0.00	0.00	cart3d::init(int)
0.00	771.31	0.01	4	0.00	0.00	diffeq::init(int)
0.00	771.32	0.01	4	0.00	0.61	modeldata::spin_galaxy(double)
0.00	771.32	0.00	1040384	0.00	0.00	modeldata::comp_s_rad(double, double)
0.00	771.32	0.00	512	0.00	0.00	mapPoints(int, int, double*, double*, cart3d*, cart3d*, cart3d*, cart3d*)
0.00	771.32	0.00	504	0.00	0.60	run_step()
0.00	771.32	0.00	18	0.00	0.00	MPI::Is_initialized()
0.00	771.32	0.00	18	0.00	0.00	MPI::Comm::~Comm()
0.00	771.32	0.00	18	0.00	0.00	MPI::Comm_Null::~Comm_Null()
0.00	771.32	0.00	9	0.00	0.00	global constructors keyed to g_dynamic

Simplified a bit

- Still need to learn how to parallelize code

Thank You

Questions?