Fractals

Outcomes

Example results

Example Results for a varying image size (-S flag) and number of cores and iterations (-i flag): 5000:

Application Time	(seconds)			
Cores / Image Size	1000	2000	4000	8000
4	3.18	10.80	41.58	164.33
8	2.00	6.29	23.04	89.90
16	1.60	4.19	14.55	56.44
32	1.58	3.47	11.02	40.72

Table 2: Example Full Applications Times for Core and Image Sizes

Calculation Time	(seconds)			
Cores / Image Size	1000	2000	4000	8000
4	2.19	8.75	35.04	140.05
8	1.00	3.98	15.92	63.66
16	0.49	1.91	7.50	29.97
32	0.25	0.97	3.67	14.51

Table 3: Example Calculation Times for Core and Image Sizes

Key points to take away

TASK FARMS

- Also known as the master/worker pattern
- Allows a master process to distribute work to a set of workers processes.
- Can be used for other types of tasks but it complicates the situation and other patterns may be more suitable for implementing.
- Master process is responsible for creating, distributing and gather the individual jobs.

Key points to take away

TASKS

- Units of work
- Vary in size, do not have to be of consistent execution time. If execution times are known it can help with load balancing. For a task farm, the tasks should as far as possible be discrete units of work from the overall

QUEUES

- Master generates a pool of tasks and puts them in a queue
- Workers assigned task from queue when idle

Key points to take away

LOAD BALANCING

- How a system determines how work or tasks are distributed across workers (processes or threads)
- Successful load balancing avoids idle processes and overloading single cores
- Poor load balancing leads to under-utilised cores, reducing performance.

Key points to take away

COST

- Increasingly important
- Finite budgets require optimal use of resources requested.
- Load balancing is just one method of ensuring optimal usage and avoiding wasting resources.
- More power and resources do not necessarily mean improved performance.
- Always ask – is it necessary to run this on 4000 cores or could it be run on 2000 more efficiently?