Vampire Numbers - Distributed Operating Systems - Project 1

Nicholas Kroeger

Marco Pagani

Our program will, by default, distribute the supplied range of values dynamically between processes. Each process receives 1000 numbers to evaluate, with as many processes as needed being created to calculate the entire range.

Optionally, you may supply a third argument at the command line, which will change the size of the heap given to each process, resulting in a different number of processes each time. Also supplying 0 as the heap size will divide the range into 8 equal heaps resulting in 8 processes exactly.

**Program output example:**

$ time mix run proj1.exs 100000 200000

Finding Numbers...

102510 201 510

104260 260 401

105210 210 501

105264 204 516

105750 150 705

108135 135 801

110758 158 701

115672 152 761

116725 161 725

117067 167 701

118440 141 840

120600 201 600

123354 231 534

124483 281 443

125248 152 824

125433 231 543

125460 246 510 204 615

125500 251 500

126027 201 627

126846 261 486

129640 140 926

129775 179 725

131242 311 422

132430 323 410

133245 315 423

134725 317 425

135828 231 588

135837 351 387

136525 215 635

136948 146 938

140350 350 401

145314 351 414

146137 317 461

146952 156 942

150300 300 501

152608 251 608

152685 261 585

153436 356 431

156240 240 651

156289 269 581

156915 165 951

162976 176 926

163944 396 414

172822 221 782

173250 231 750

174370 371 470

175329 231 759

180225 225 801

180297 201 897

182250 225 810

182650 281 650

186624 216 864

190260 210 906

192150 210 915

193257 327 591

193945 395 491

197725 275 719

real 0m37.443s

user 2m6.519s

sys 0m1.736s

Ratio of CPU to Real time:

**CPU time** = **User time** + **Sys time**

**User** time = 2m6.519s = 126.519s

**Sys** time = 1.736s

**CPU time** = 128.255

**CPU**/**Real** = 128.255 / 37.443s = **3.419**

We attempted to find vampire numbers between **100,00 and 1,000,000**. The results are below

$ time mix run proj1.exs 100000 1000000

102510 201 510

104260 260 401

105210 210 501

105264 204 516

105750 150 705

108135 135 801

110758 158 701

115672 152 761

116725 161 725

117067 167 701

118440 141 840

120600 201 600

123354 231 534

124483 281 443

125248 152 824

125433 231 543

125460 246 510 204 615

125500 251 500

126027 201 627

126846 261 486

129640 140 926

129775 179 725

131242 311 422

132430 323 410

133245 315 423

134725 317 425

135828 231 588

135837 351 387

136525 215 635

136948 146 938

140350 350 401

145314 351 414

146137 317 461

146952 156 942

150300 300 501

152608 251 608

152685 261 585

153436 356 431

156240 240 651

156289 269 581

156915 165 951

162976 176 926

163944 396 414

172822 221 782

173250 231 750

174370 371 470

175329 231 759

180225 225 801

180297 201 897

182250 225 810

182650 281 650

186624 216 864

190260 210 906

192150 210 915

193257 327 591

193945 395 491

197725 275 719

201852 252 801

205785 255 807

211896 216 981

213466 341 626

215860 251 860

216733 323 671

217638 321 678

218488 248 881

226498 269 842

226872 276 822

229648 248 926

233896 338 692

241564 461 524

245182 422 581

251896 296 851

253750 350 725

254740 470 542

260338 323 806

262984 284 926

263074 437 602

284598 489 582

284760 420 678

286416 468 612

296320 320 926

304717 431 707

312475 431 725

312975 321 975

315594 534 591

315900 351 900

319059 351 909

319536 336 951

326452 524 623

329346 342 963

329656 356 926

336550 530 635

336960 360 936

338296 392 863

341653 533 641

346968 366 948

361989 369 981

362992 392 926

365638 533 686

368550 585 630

369189 381 969

371893 383 971

378400 473 800

378418 431 878

378450 435 870

384912 432 891

386415 465 831

392566 593 662

404968 446 908

414895 491 845

416650 641 650

416988 468 891

428980 482 890

429664 464 926

447916 476 941

456840 540 846

457600 650 704

458640 546 840

475380 570 834

486720 624 780

489159 549 891

489955 545 899

498550 590 845

516879 681 759

529672 572 926

536539 563 953

538650 630 855

559188 588 951

567648 657 864

568750 650 875

629680 680 926

638950 650 983

673920 720 936

679500 750 906

729688 788 926

736695 765 963

738468 843 876

769792 776 992

789250 875 902

789525 825 957

792585 855 927

794088 807 984

809919 891 909

809964 894 906

815958 858 951

829696 896 926

841995 891 945

939658 953 986

real 5m46.045s

user 20m39.632s

sys 0m11.814s

Below are two screenshots of the windows resource monitor, showing a single process run vs a 12-process run on a system with 12 logical threads. In the first, the process is passed between cores by the erlang scheduler, but only one is utilized at a time. In the second, all cores are operating simultaneously to process the values over a much shorter timespan.

