Alperen Degirmenci
CS 205 - HW 1
09/26/2012

Problem 2

$P2a.py$ returned $\frac{\pi}{4} = 0.7853981541006446$, and $P2b.p$ returned $\frac{\pi}{4} = 0.7853981541006307$. It looks like the difference of $139 * 10^{-16}$ is caused by error accumulation due to machine precision (double precision is $10^{-}16$).

	Local Machine	Resonance Node	Amazon EMR
P2a.py	9.79141807556	4.95454406738	429.13776803
P2b.py	8.29986810684	2.08260703087	399.001466036

Table 1: Running times (in seconds) of MapReduce jobs with and without in-mapper combining on Local Machine, Resonance Node, and Amazon EMR.

In all cases, the in-mapper combining made the MapReduce run faster (only the MR job portion of the code was timed). The speedup is most pronounced on the resonance node, where the MR job completed in less than half the time with in-mapper combining.

I have actually written more complicated version where the bounds of the x-axis can be set by the user. Input is then generated depending on the range. Input/output redirection is handled by the program and pipelining is not necessary (and actually not supported). I wrote this before I realized that we were supposed to submit a single .py file for each implementation.