• HW0 parts 4-6 due today.
• git tutorial repeat Monday at 3, this room.
• HW1 should be ready Monday (Spark).
• Walkthroughs for installing Spark locally, and using it on AWS - on course webpage soon.
Colloquium
Causes of poor speedup in parallel algorithms

- insufficient parallelism
- scheduling overhead
- memory bandwidth
- contention (locking and true/false sharing)
• Denard scaling
  • power density stays constant (until 2005)
• Amdahl’s law
  • If 1/N of your program is serial, maximum speedup is N
• span vs. work:
  • work = total instructions = $T_1$
  • span = critical path instructions = $T_∞$
  • parallelism = $T_1 / T_∞$
• Death of Moore’s law:
  • Current 14nm, 140 angstroms
  • physical minimum = 5nm = 3 years
• Intel Development process
  • tick/tock = process (nm) / architecture
  • tick/tock/refresh = process (nm) / architecture / architecture
• Google: “intel tick tock wiki”
• python performance
  • optimized 4K x 4K matrix multiply down to... 1/2 second?
  • Python = 7 hours
  • try numpy?

• cilk vs. more fine-grained or user-tunable systems
  • argument: cilk is a tradeoff between simplicity and control.
Spark (continued)
Spark Recap

RDDs, Transformations, Actions
Resilient Distributed Datasets

- **Immutable.**
- Created from raw data or Transformations of other RDDs.
- Partitioned & distributed across workers.
- Know their history (can be recomputed).
- Cachable.
Transformations

• Turn one or more RDD into another RDD.
  
  map
  filter
  sample
  union
  intersection
  groupByKey
  reduceByKey
  join
  cache

  FooByKey operates on (K,V) tuples.
Actions

- Turn an RDD into a result passed back to the caller.
  
  `reduce`
  `collect`
  `count`
  `first`
  `take(n)`
  `saveAs...`
  `countByKey`
Example: Word Count

“Hello World” for analytics platforms.

• Load lines of text
Example: Word Count

“Hello World” for analytics platforms.

- Load lines of text
- Map to split into words
- Map to convert to \(<K, V>: (word, 1)\)
- ReduceByKey to get per-word counts.
- TakeOrdered to get largest counts.
Example: Suffix Words

• Find all words in a dictionary where removing the first letter still gives a word.

• … load lines of text …
Shuffling

A stage = sequence of operations without a shuffle.
Shuffles in Word Count

• Load lines of text
• Map to split into words
• Map to convert to \(<K, V>: (word, 1)\)
• ReduceByKey to get per-word counts.
• TakeOrdered to get largest counts.
Suffix Words

(word, word)
(word, word)
(suffix, word)
(suffix, word)
(suffix, word)
(suffix, word)
(suffix, word, word)
(suffix, word, word)
(suffix, word, word)
Shuffling can be Slow

- Be cognizant of how much data is generated in a shuffle.
- Large shuffles will end up spilling to disk and use lots of network bandwidth.
Word count revisited

• reduceByKey() vs groupByKey()
Questions?