On the Hunt for Correctness and Performance Bugs in Large-scale Programs

Milind Kulkarni, Saurabh Bagchi and Michael Gribskov
Purdue University

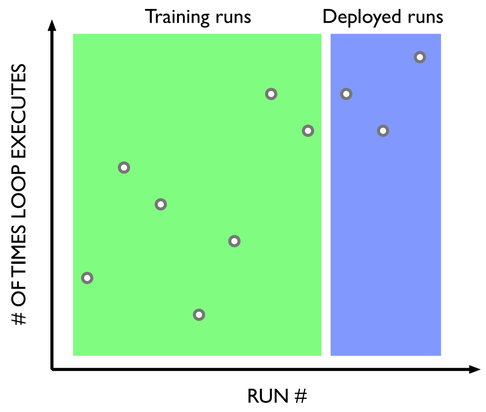
Scaling issues in programs

- System scale is increasing dramatically
  - Larger machines
  - Larger inputs
- Larger scales bring scalability issues:
  - Bottlenecks that prevent scaling (synchronization, communication, etc.)
  - Bugs that arise due to scaling up (races, overflows, etc.)
- Detecting, diagnosing and fixing scaling issues is complex and challenging
- This project investigates (semi) automatic approaches for detecting, diagnosing and fixing scaling issues, with a special emphasis on computational genomics applications

Automatically Detecting and Localizing Bugs that Manifest at Large System Scales

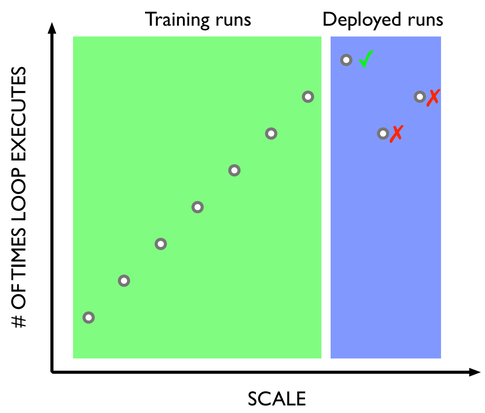
- Statistical debugging technique for detecting bugs at large system scales
  - Key idea behind statistical debugging: build a model of correct program behavior, flag deviations from that model as bugs
  - Approach has issues when scaling up programs: even normal program behavior changes with program scale!

Is there a bug in one of the deployed runs?



- WuKong builds *scaling models* of programs, relating system/input scale to program behavior
  - Train at many small scales to build model that relates *control features* (scale) to *observational features* (program behavior)
- During deployment:
  - Deviation from scaling model → bug
  - Deviant feature → likely bug location.
- Detect bugs at deployed scales—even if never trained on correct behavior at large scale!

Accounting for scale makes trends clear, errors at large scales obvious

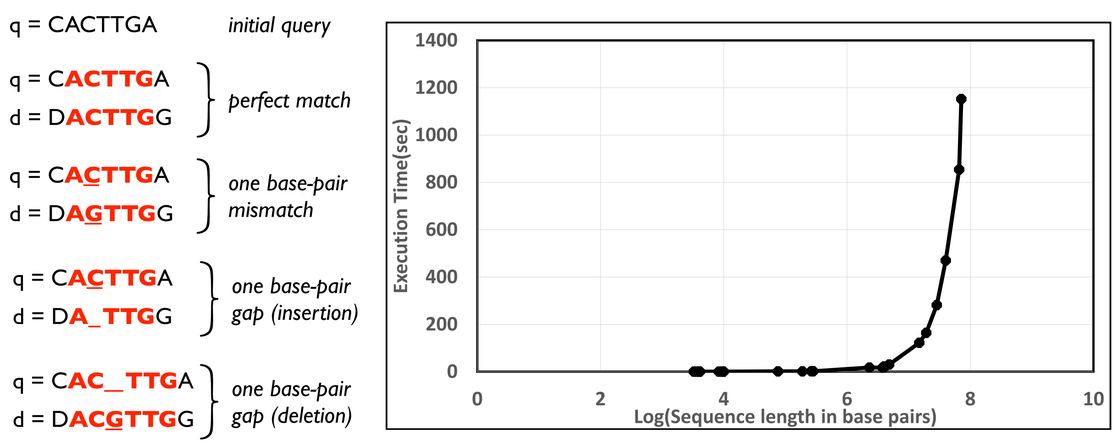


Bowen Zhou, Jonathan Too, Milind Kulkarni and Saurabh Bagchi, "WuKong: Automatically Detecting and Localizing Bugs that Manifest at Large System Scales" HPDC 2013.

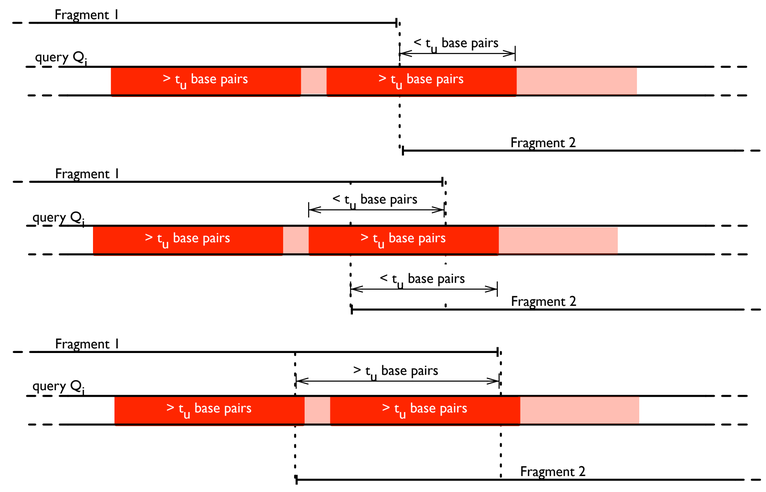
Scaling Up Sequence Alignment

- Sequence alignment (finding overlapping sequences) is a key kernel in computational genomics
  - Can be nucleotide sequences or amino acid sequences
  - Matches do not need to be exact
- BLAST (Basic Local Alignment Search Tool) is the state-of-the-art alignment tool
- mpiBLAST is state-of-the-art parallel version: aligns *query* sequences against databases of *reference* sequences

  Has scalability bottleneck: if sequences are long, mpiBLAST runs out of memory

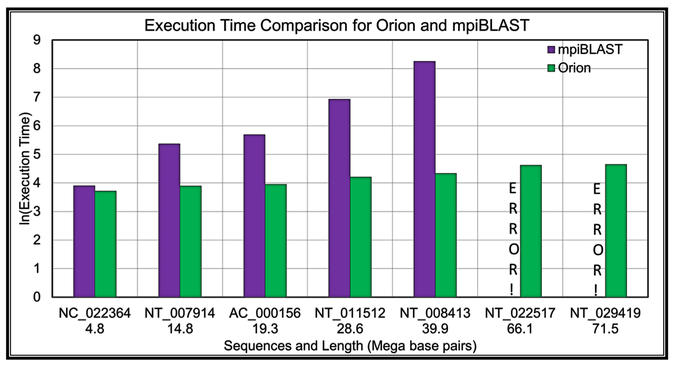


- Orion exploits a finer granularity of parallelism, *intra-query parallelism:*
  1. Partition queries into *fragments*
  2. Fragments must overlap to avoid missing alignments!

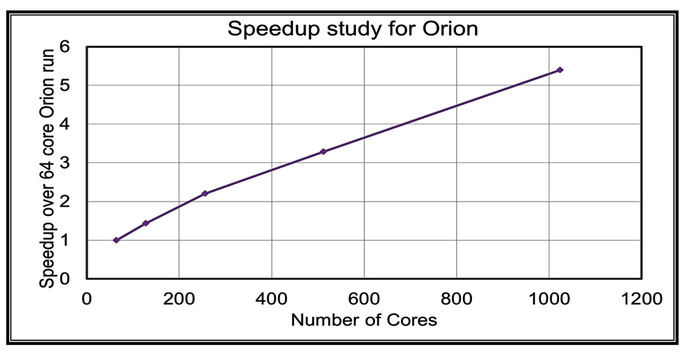


  3. Perform alignment on each fragment
  4. Merge together partial alignments to produce final result

Comparison of Orion and mpiBLAST for different query lengths. Note that mpiBLAST runs out of memory for the longest queries



Orion scalability running on a cluster of 64 16-core nodes, normalized to speedup on 64 cores.



Kanak Mahadik, Somali Chaterji, Bowen Zhou,, Milind Kulkarni and Saurabh Bagchi, "Orion: Scaling Genomic Sequence Matching with Fine-Grained Parallelization" Supercomputing 2014.