

EuroHack 2018
Lugano – October 2018

GPU-Aevol – 1.0; 2.0; 3.0

Guillaume Beslon – Computational Biology

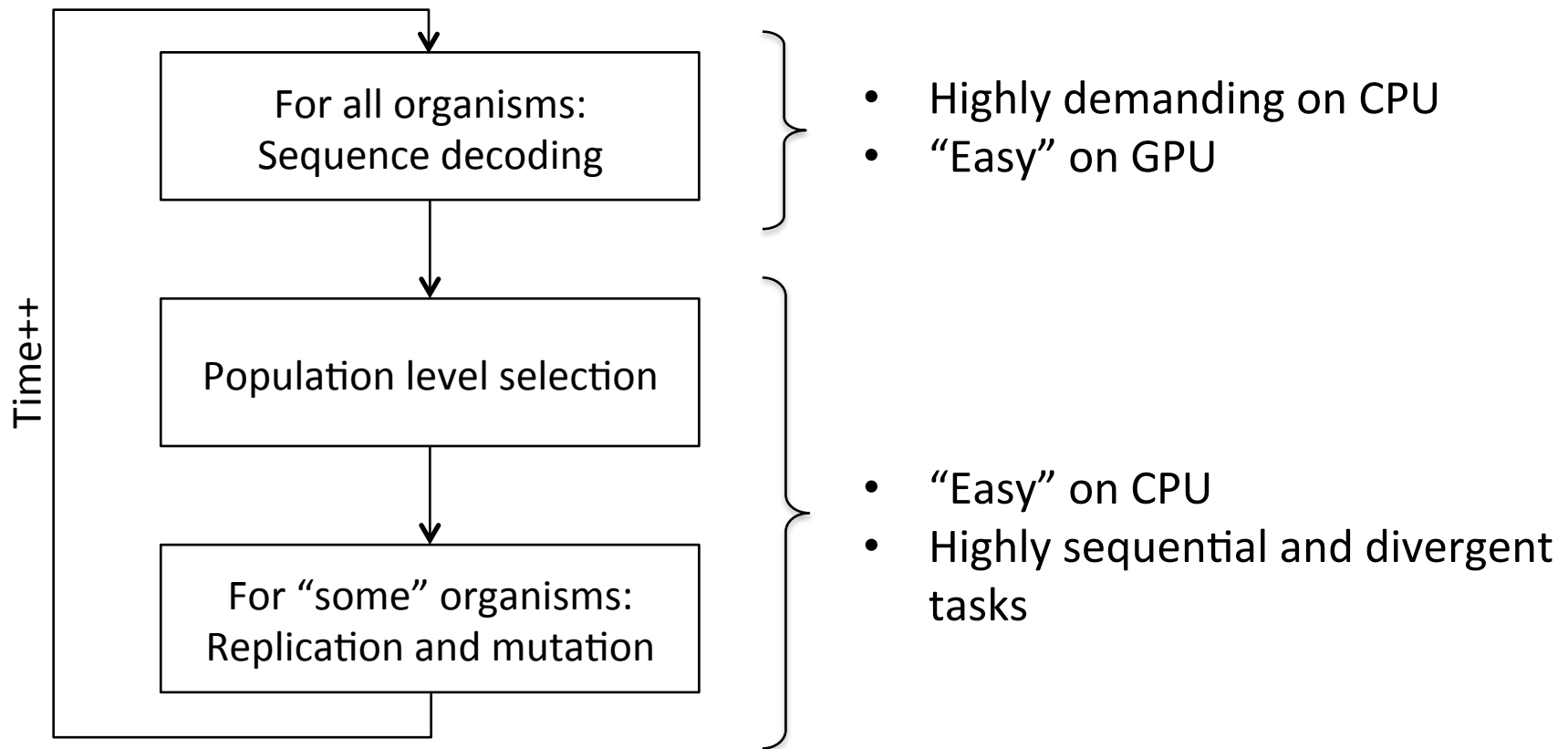
David P. Parsons – Software engineering

Jonathan Rouzaud-Cornabas – High Performance Computing

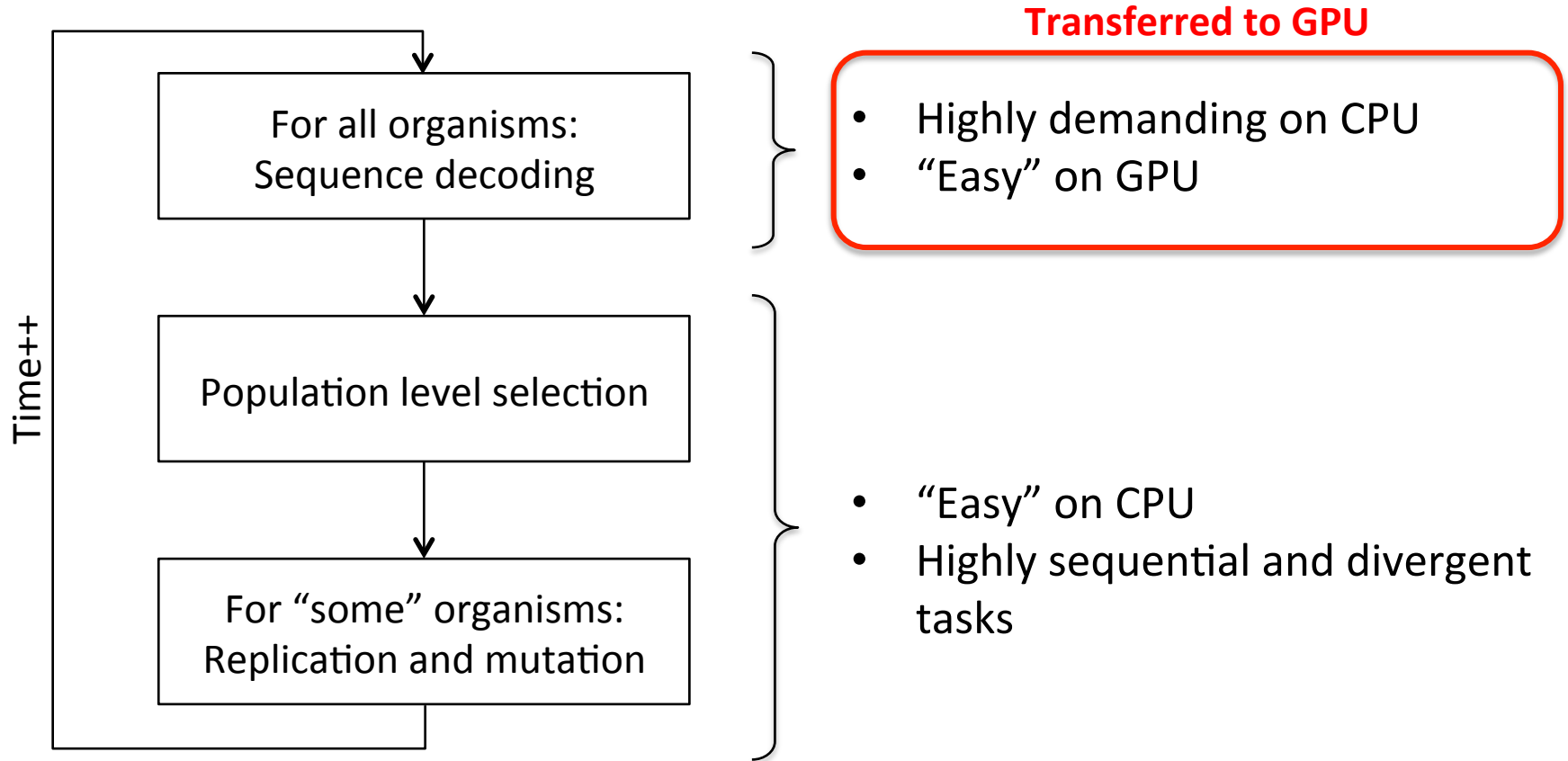
Mentors: Vasileios Karakasis (ETH Zurich), Jeffrey Kelling (HZDR, Dresden)



A “closer” look at Aevol architecture



GPU-Aevol V1

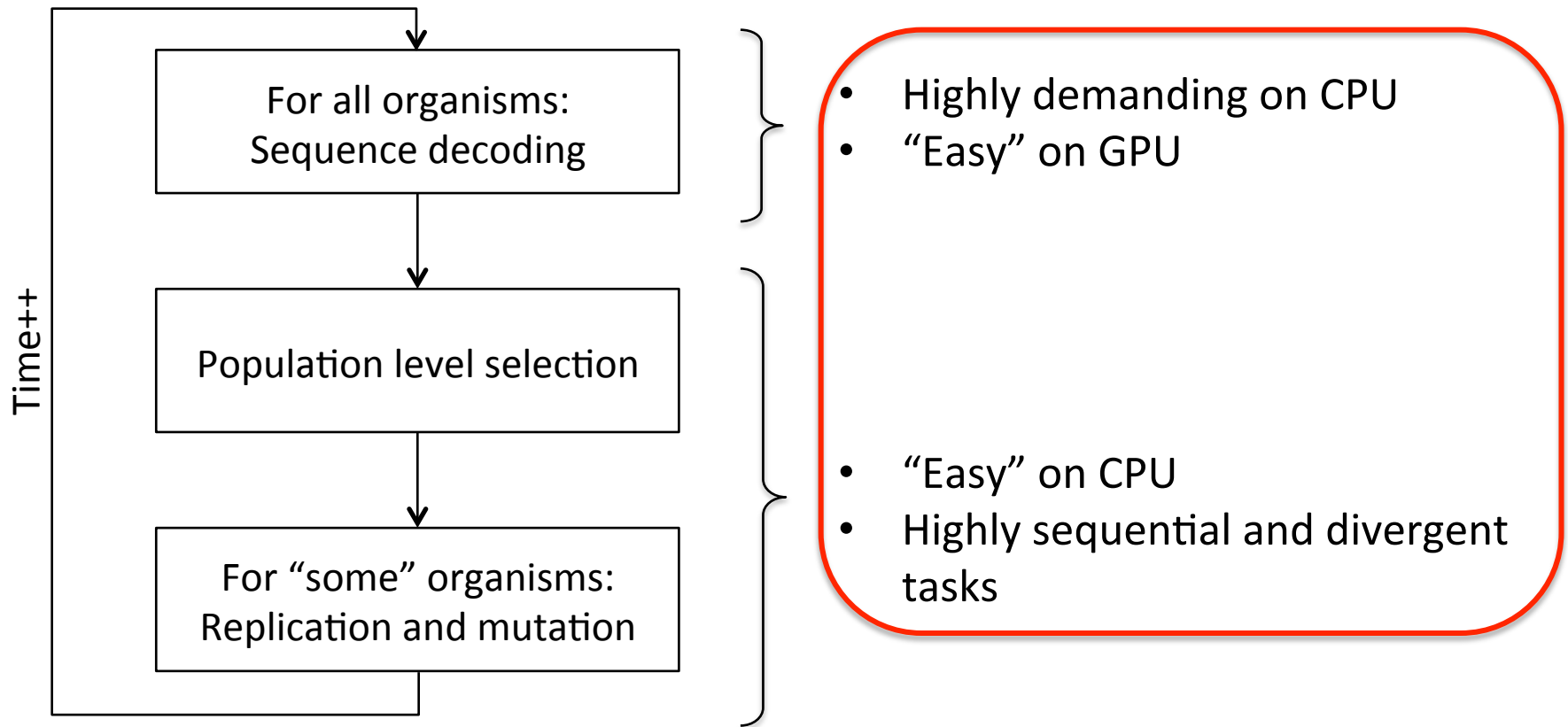


GPU-Aevol V1

(Tuesday evening)

- Sequence decoding on GPU
 - Not so easy due to long-range dependencies on DNA (start-stop signals may be found by different threads)
 - 2-steps decoding (finding signals, associating signals one to the other)
 - Next decoding steps are less demanding have been efficiently ported on GPU on Wednesday morning
- GPU-Aevol V1:
 - Full decoding on GPU (from sequence to fitness)
 - Selection and mutations on CPU
- Profiling of GPU-Aevol V1
 - Highly inefficient compared to CPU-optimized code
 - Explanation: too many CPU-GPU transfer (long sequences)
 - Next objective: transfer the whole algorithm on GPU to avoid data transfer

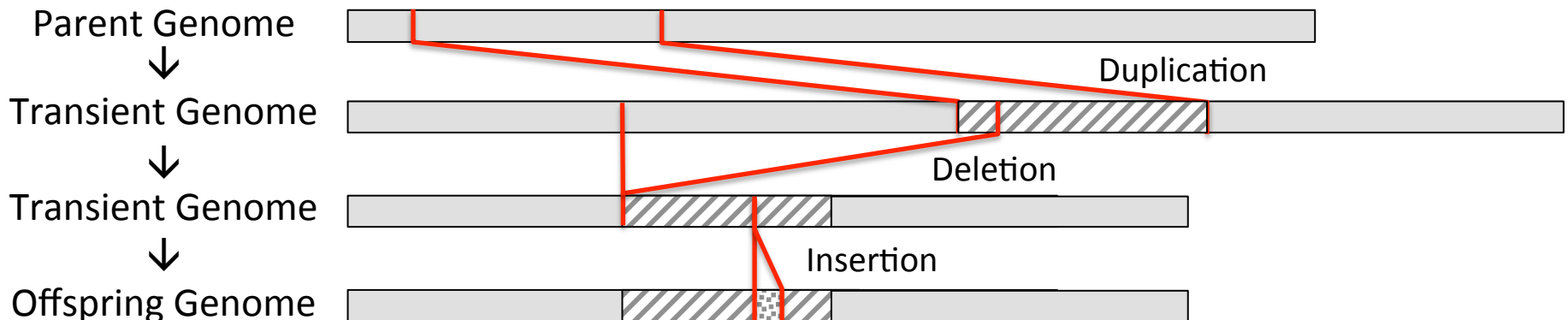
GPU-Aevol V2



GPU-Aevol V2

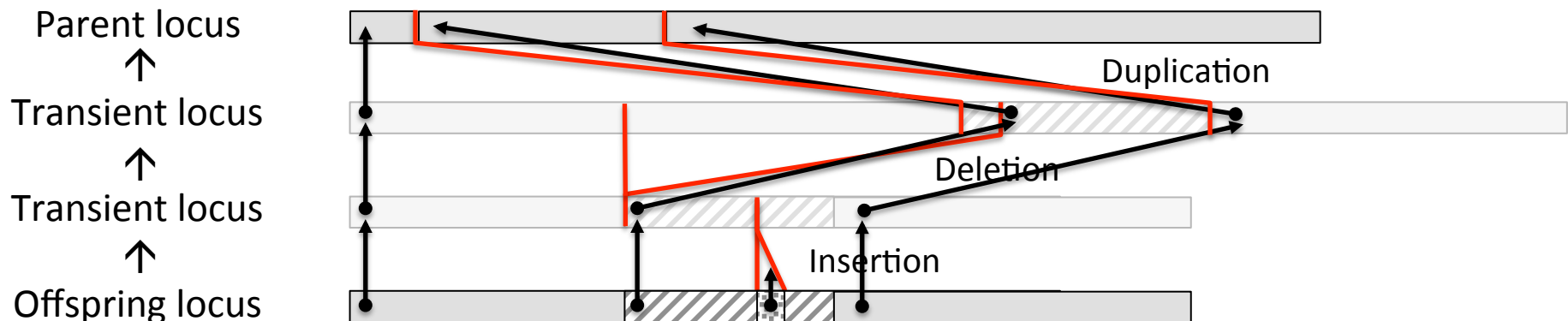
(Wednesday evening)

- Sequence decoding, selection and mutations on GPU
 - CPU-GPU data exchange only at first generation
 - GPU-Aevol V2 was running on Wednesday afternoon
- Profiling of GPU-Aevol V2
 - Awfully, Awkwardly, Badly, Definitely, unpredictably SLOW!!!
 - 97.1 % of time in mutation kernel
 - Explanation: due to structural mutation operators (~copy/paste of subsequences) mutation requires extensive memory management



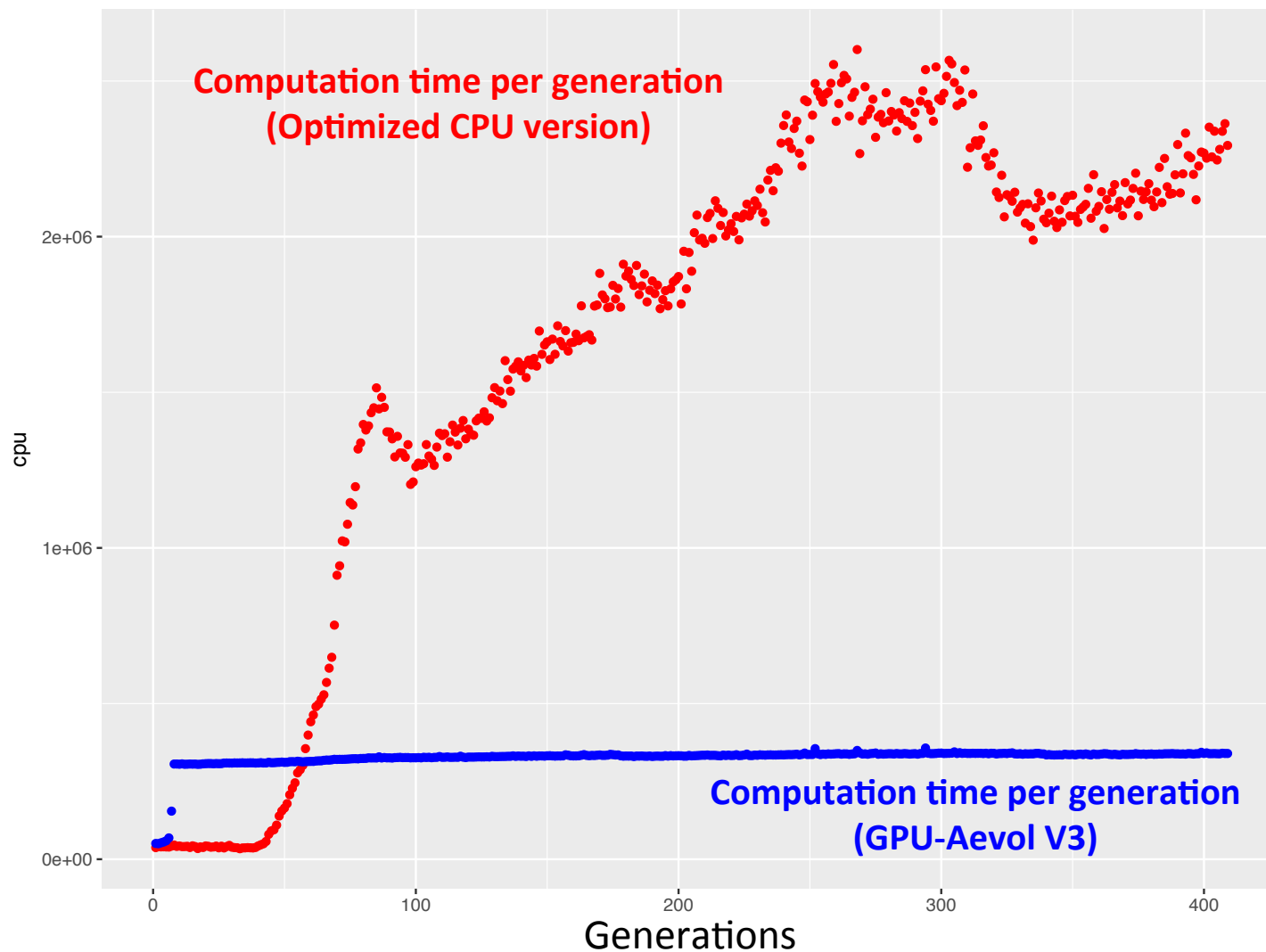
The tricky part...

- Refactoring the mutation algorithm to (1) avoid computing transient sequences (2) enable massive parallelization
- Step 1: precompute mutations
 - Only requires computing transient sequence sizes
 - Store a table of mutations for each organism
- Step 2: backtrace the nucleotides of the offspring up to the parent
 - Computation of nucleotides become fully independent

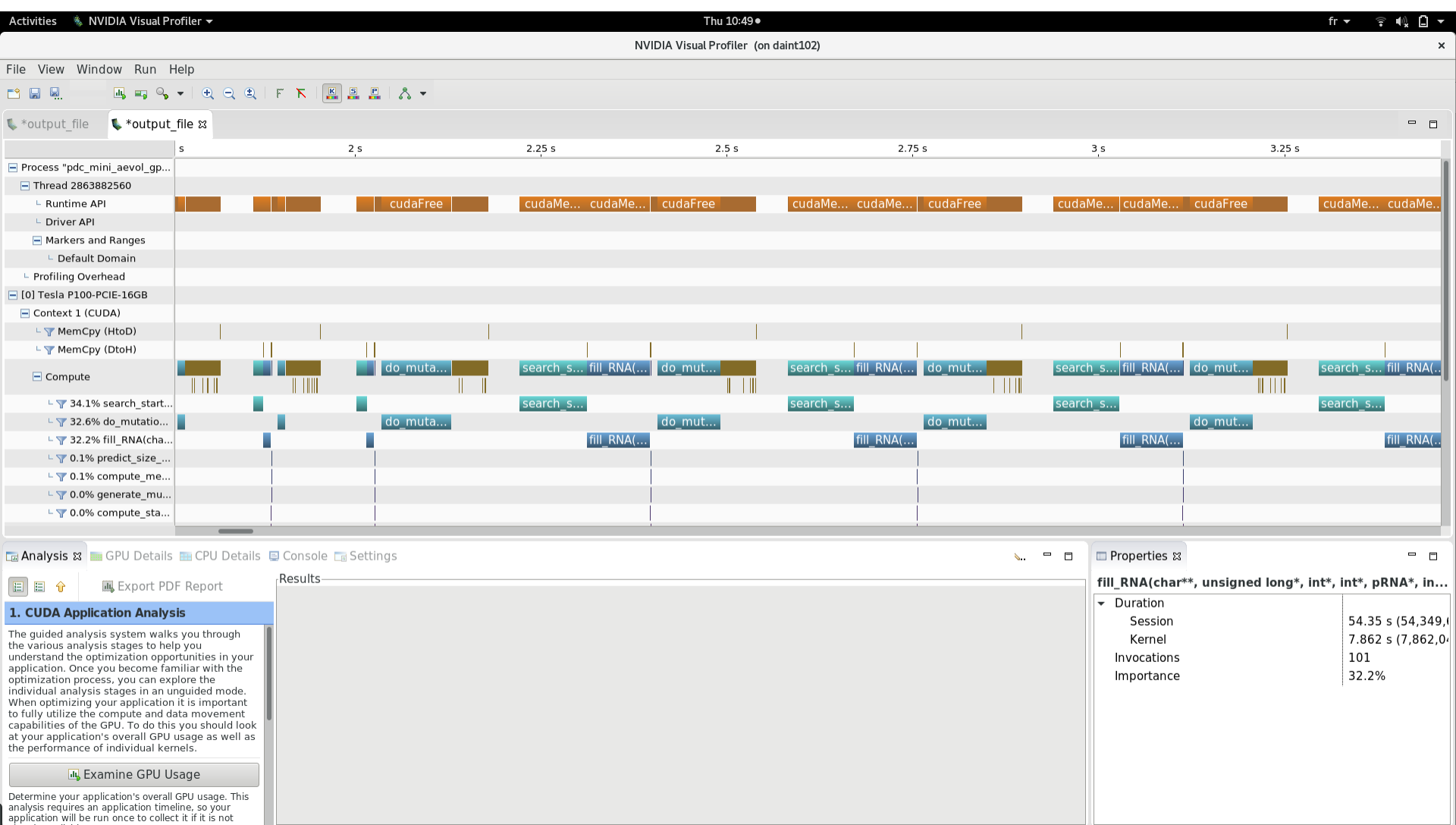


Current state: GPU-Aevol V3

(Thursday morning)



Ongoing: profiling of GPU-Aevol V3



To be done: optimizing GPU-Aevol V3

- Architecture and parallelization scheme considered (almost) stable
- Optimization ideas:
 - Assemble all genomes into a metagenome to get rid of size variations
 - Would enhance efficiency of decoding and mutation kernels
 - Fusion of decoding and mutation kernels
 - Store metainformation along the sequence (position of the coding signals)
 - Compress sequences (char to bitstring)
 - Backtrace the position of coding signals in the mutation kernel (i.e. port the CPU optimization on GPU)
 - ...