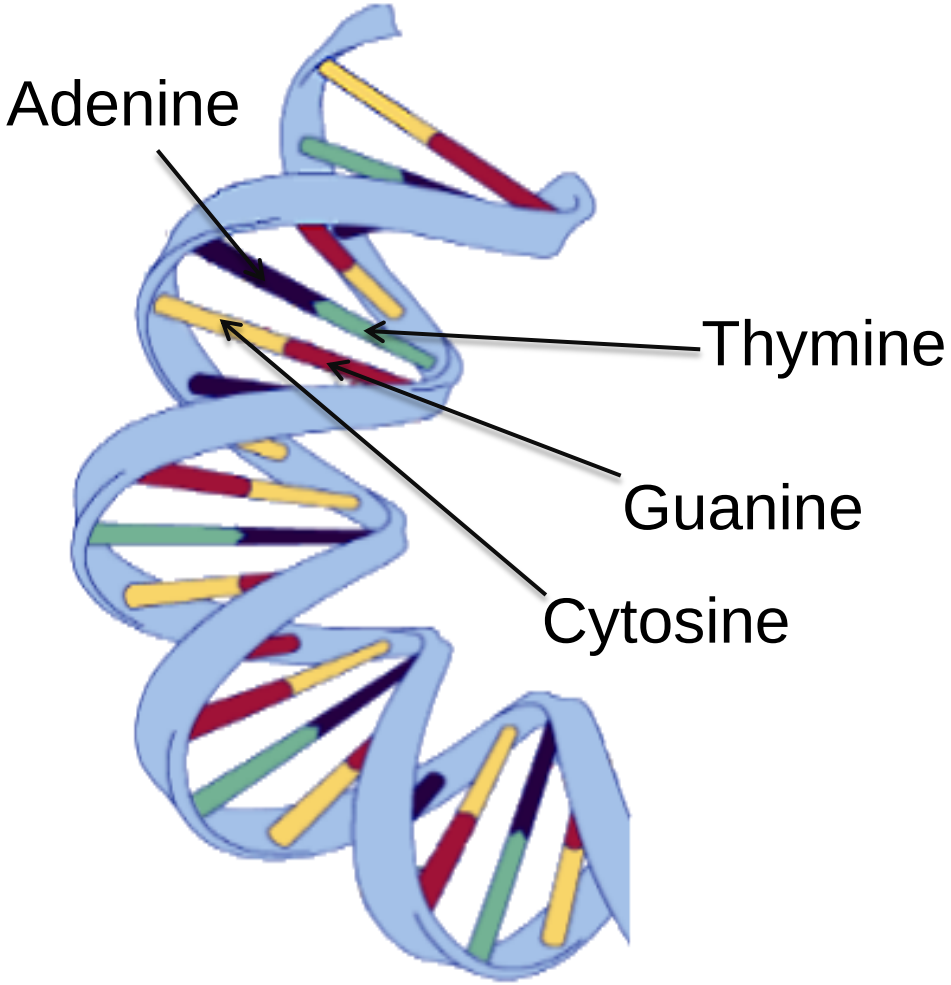
ThermoFisher SCIENTIFIC

Scaling Ion Torrent™ Semiconductor Sequencing Analysis with GPU's

Mohit Gupta and Jakob Siegel

What is DNA

- Hereditary material in humans and almost all other living organisms
- Comprises of four chemical base pairs: (A,T), (C,G) in double helix structure
- Sequence of these base pairs determines how an organism develops, survives and reproduce

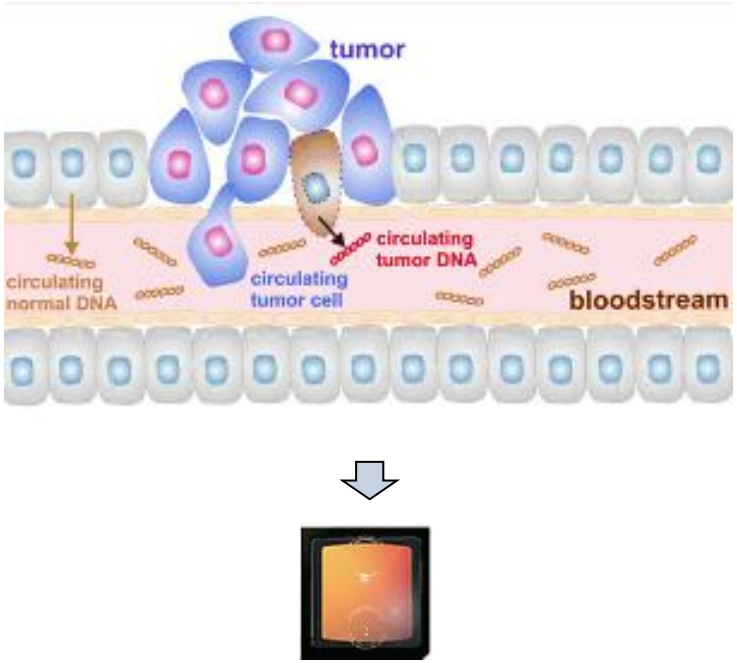


life technologies

Why sequence DNA ?

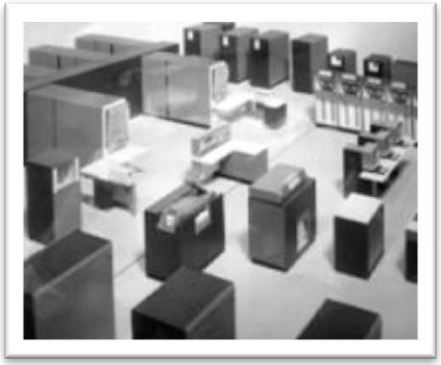
Tumor DNA: in blood

- Identifying tumor by sequencing the free floating DNA in blood for mutations
- If mutations in ctDNA → tumor somewhere in the body



Benchtop High Throughput Sequencing Accessible to All Labs

Main Frame



Mini Computer

Personal Computer

CE/Sanger Sequencing

Next-Gen Sequencing

Ion Semiconductor Sequencing

<u>Requirements for Success</u>
Affordable, Scalable Technologies
Simple, End-to-End Workflows
Strong User Community and Reference Base

life technologies

Ion Proton™ sequencer

Innovative semiconductor sequencing chip

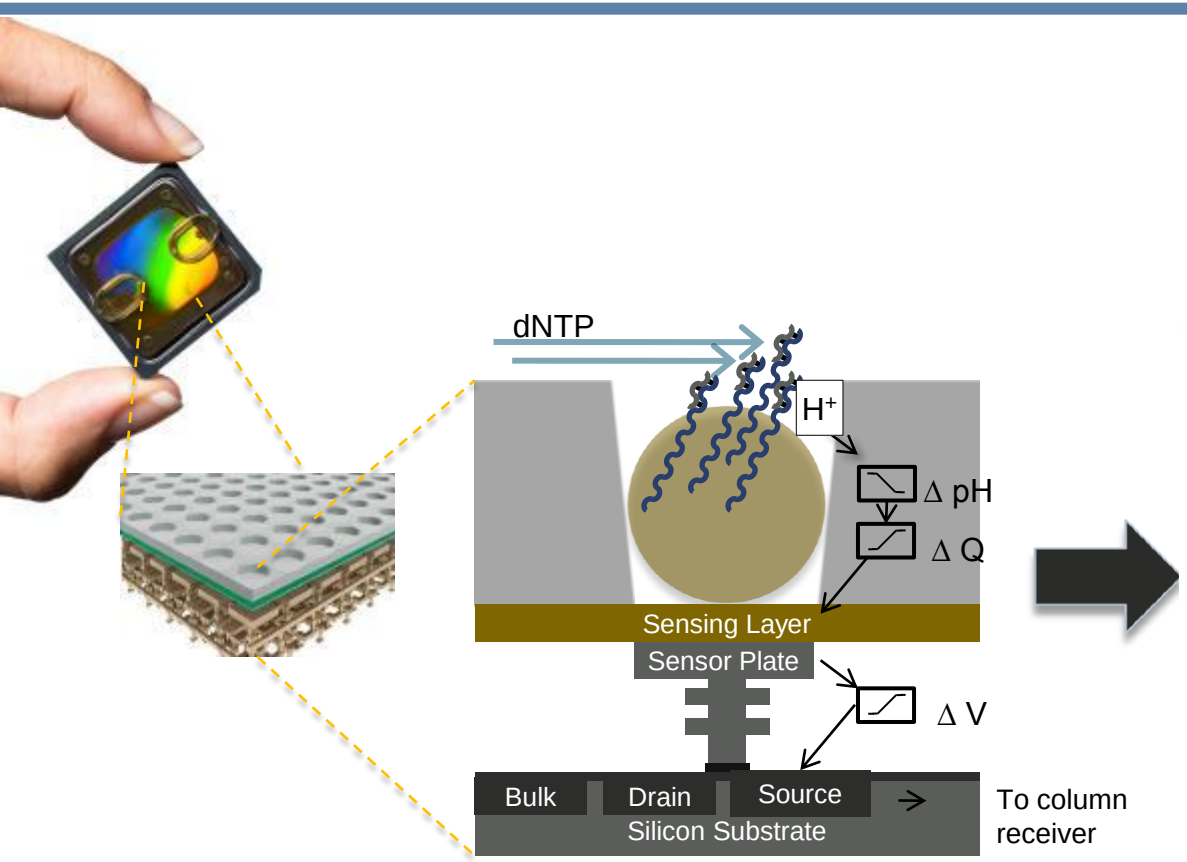
Simple chip loading

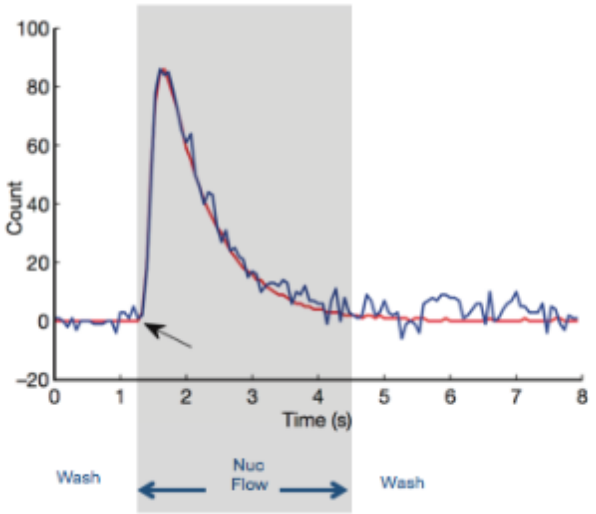
Intuitive graphical user interface

State-of-the-art electronics

Integrated reagent delivery

Transistor as a pH meter



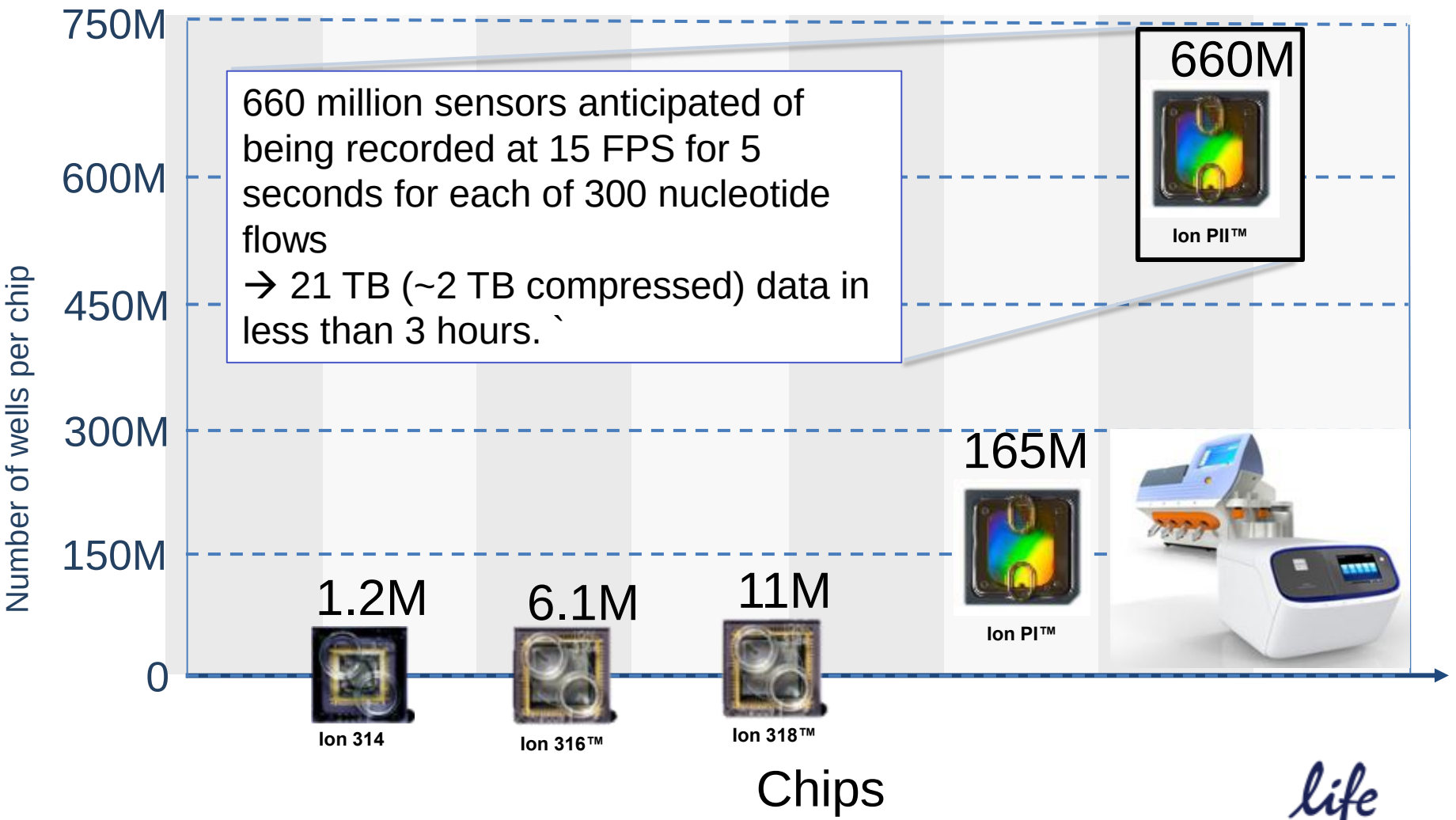


Rothberg J.M. *et al Nature doi:10.1038/nature10242*

Compute Intensive signal processing

life technologies

Unprecedented Scaling



Ion Chip Scalability

3-series, 3um	PI™, 1.3um	PII™, 0.5um
11M wells	165M wells	660M wells

Ion PII™ Chip

Contrast to the Ion PI™ Chip:

- 4x as many wells and transistors is expected to enable much higher throughput
- Expected to offer twice the data rate to maximize data analysis rates and minimize analysis times
- Anticipating higher signal to noise to maximize quality of signal output

Anticipated data rate from the PII™ Chip

Latest generation CMOS camera
5.6 Gbps

1G Ethernet
120 cables

Ion PII™ Chip
660 M wells
120 Gbps

Latest generation Cisco router
10 to100 Gbps

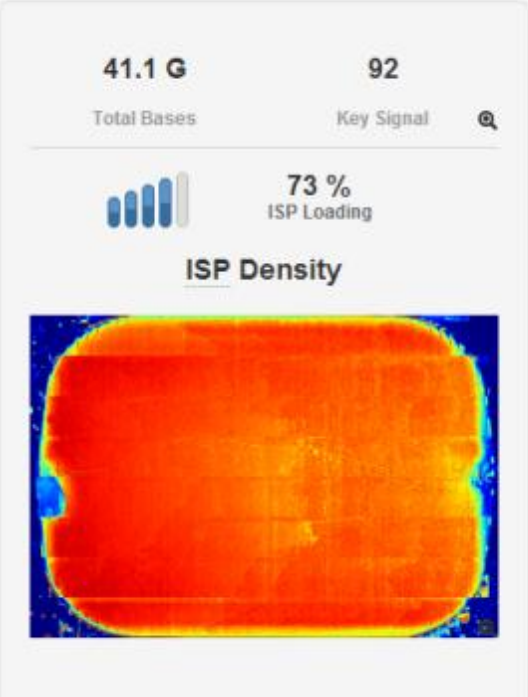
Netflix

HD Movie streams
36000 movies

Ion PII™ Chip

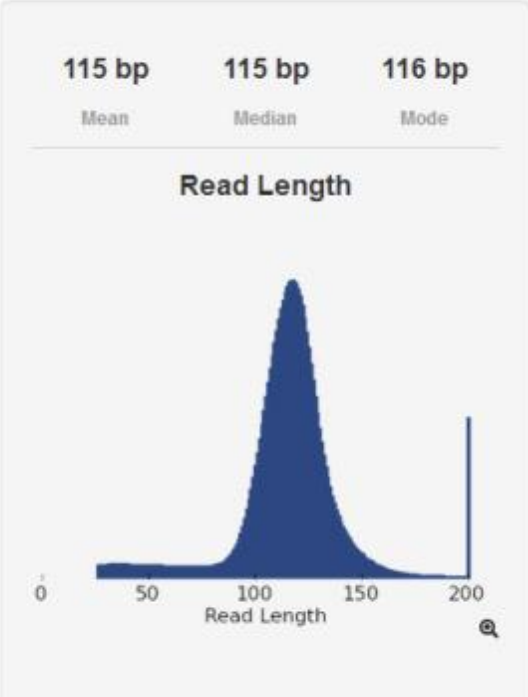
Expected to be Functional and Deliver Promising Data

>30G Total Bases



>300M aligned reads

115 base mode



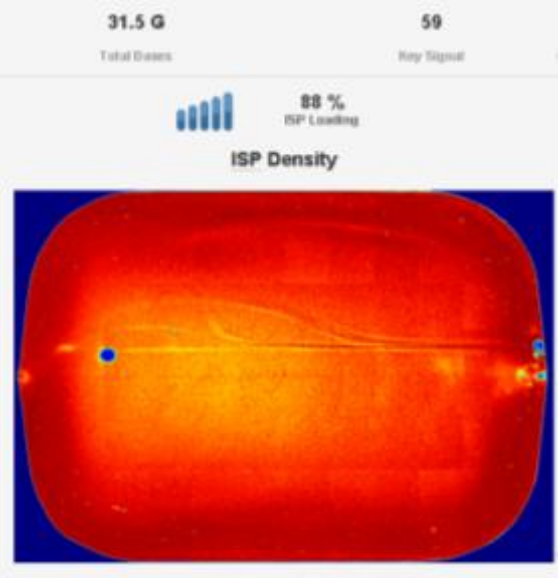
Human Genome Fragment Library

Target 200-300M reads @ 100bp

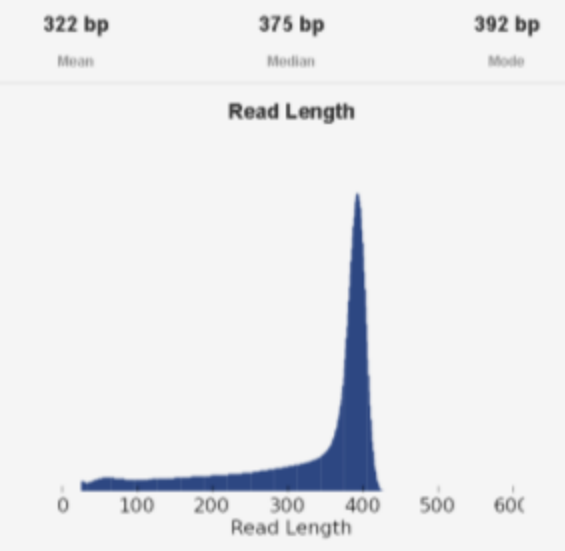
Ion PI™ Chip - Microwell Innovations

Push read length to 400 bases and output to 30Gb

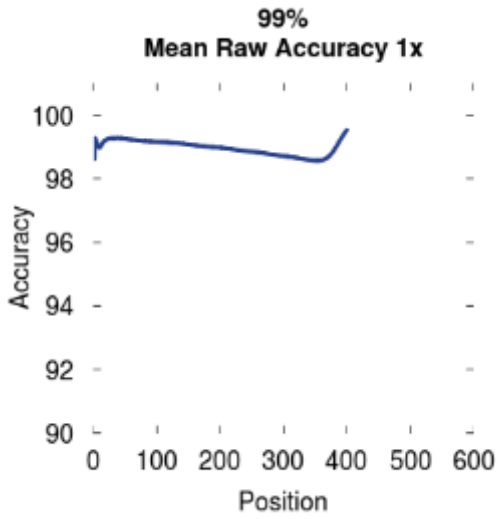
~ 30 Gb Total Bases



~ 400 Base Mode

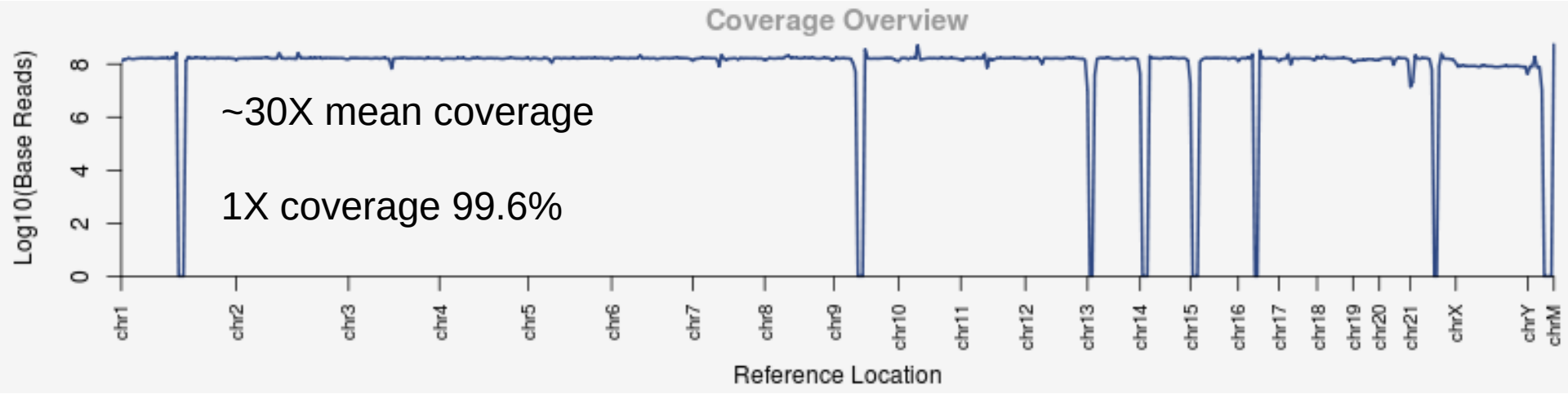


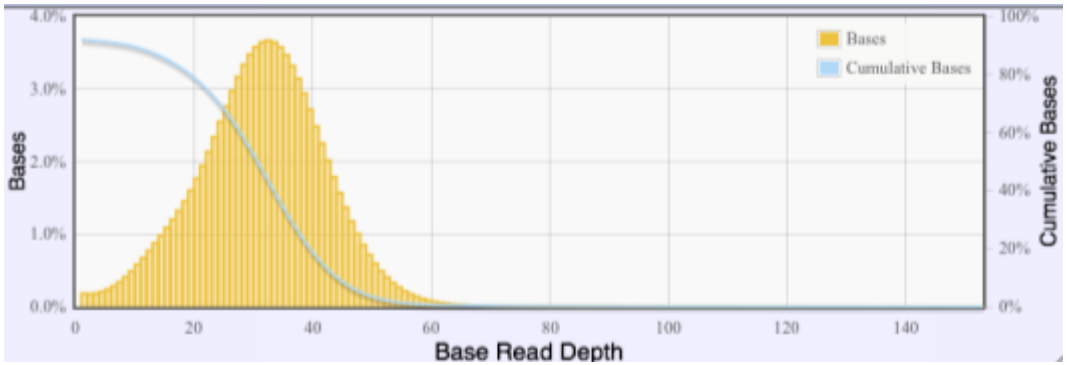
99% Accuracy



Combining 3 PI™ chips enables 30X whole human genome

30X Whole Human Genome on Three Ion PI™ Chips

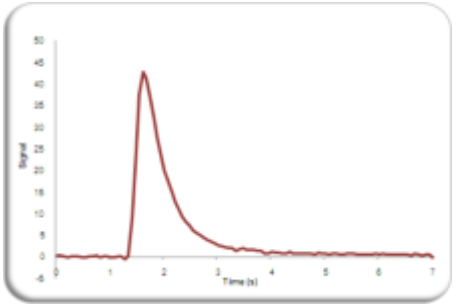




SNP:MAF>0.2	SNPs
Total	3,621,628
Het/Hom ratio	1.693
Ts/Tv ratio	2.06
% in(chr1) dbSNP	97.4%

Data Processing Pipeline

21 TB
(PII™ chip)



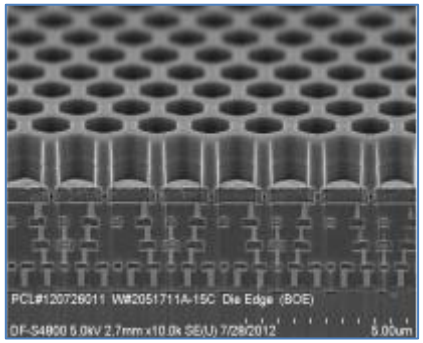
Data acquisition and compression in FPGA

2 TB

Signal Processing

200 GB

BaseCalling and Alignment to reference genome



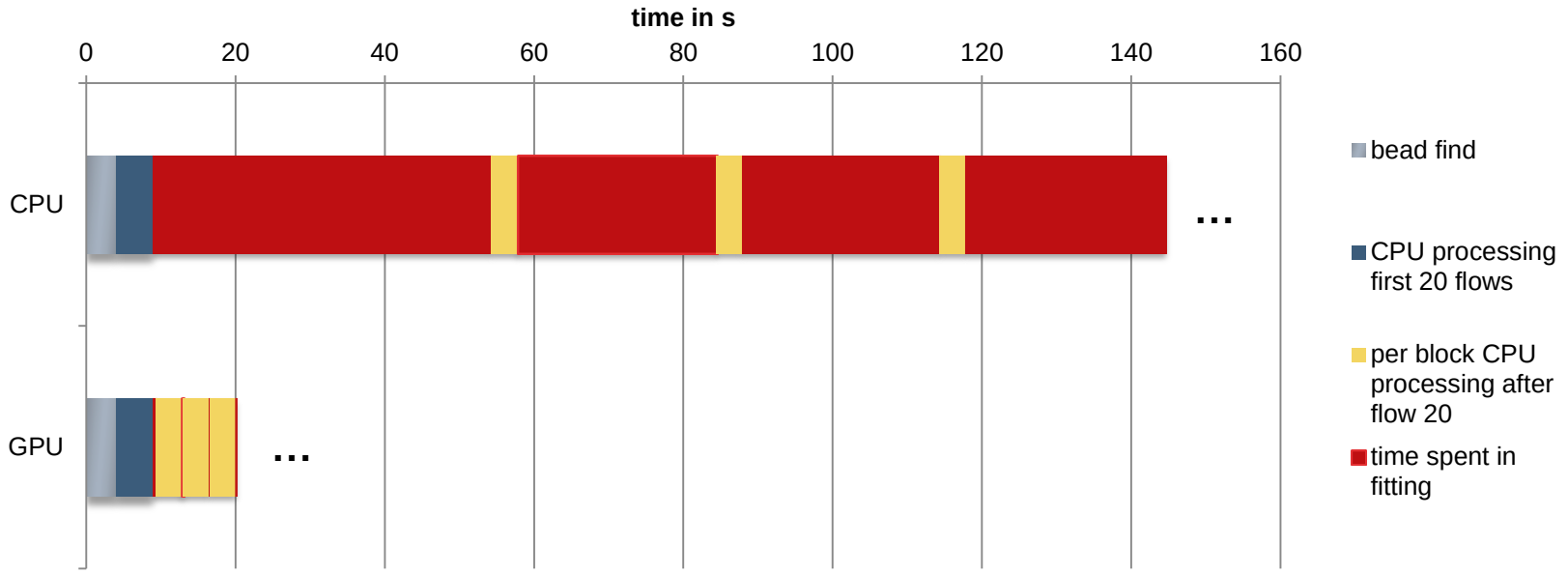
```
TGTATACCTCTTACTTTACAGATAGGTTCATTATCT
||||||||||||||||||||||||||||||||||||
TGTATACCTCTTACTTTACAGATAGGTTCATTATCT
        10        20        30
```

How to process data at the source ?

- Big data on Proton™ Sequencer
- Cloud/Cluster not an option
- Dual 8-core Intel® Xeon® Sandy Bridge
- Dual Altera® Stratix® V
- **NVIDIA® Tesla® K20**
- 11 TB (SSD and HDD)

GPU to the rescue

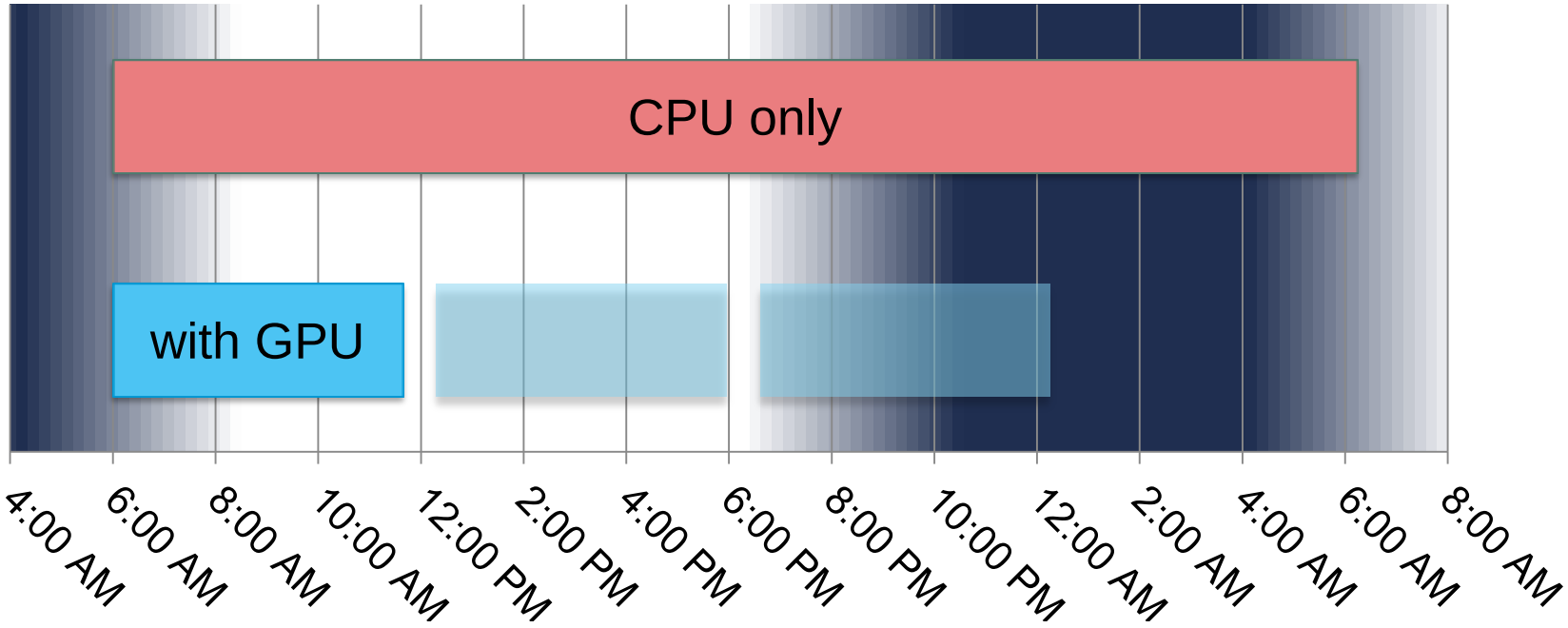
- Removed main hotspot in signal processing pipeline
- **Speedups of more than 250x over a CPU core!**



GPU's Impact

- Multiple sequencing runs a day possible
  - Swift pace of Research and Development
  - Accelerated product innovation

On Instrument Analysis Time with and without GPU



Signal Processing

Signal Processing

Signal Processing Flow

Reading flow data

Raw Data Processing

Regional Parameter Estimation (Common to all wells)

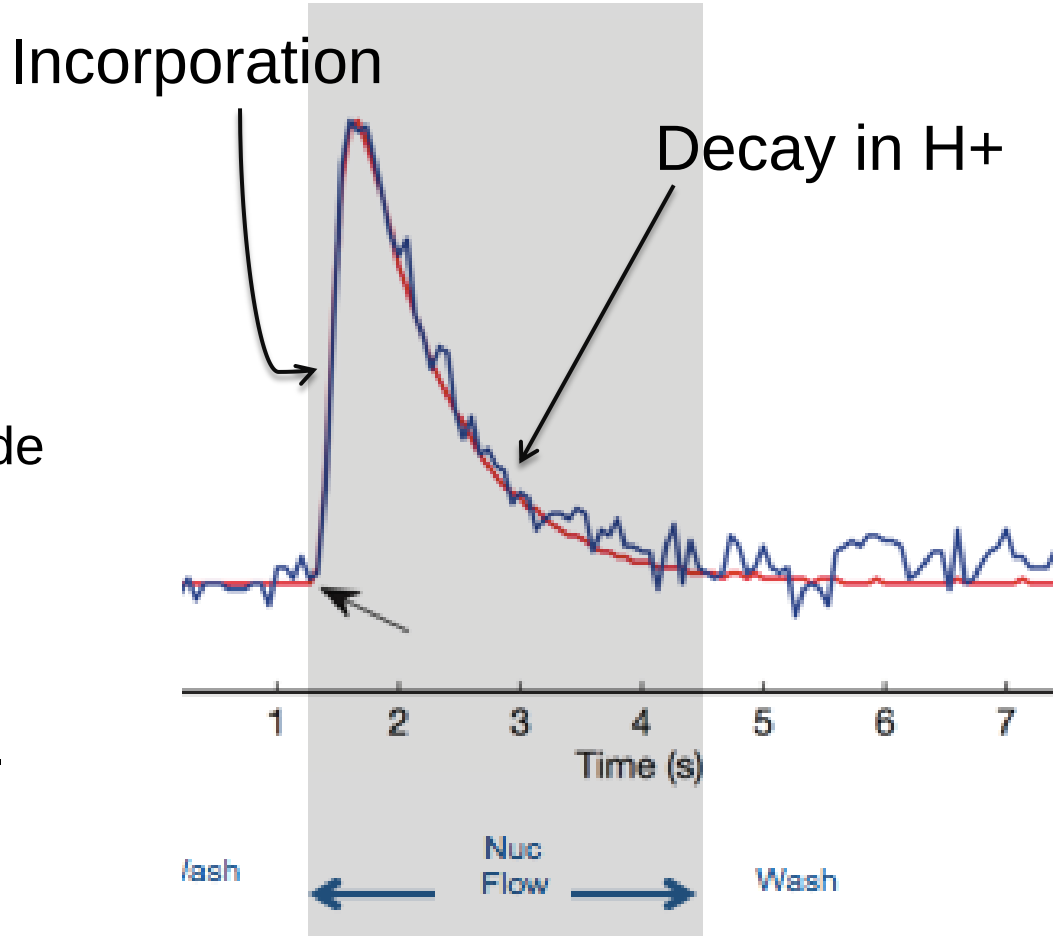
Parameter Estimation unique to each well (LM fitting)

Post Fit Processing

Writing signal values

Mathematical model

- Sophisticated model
  - Background correction
  - Incorporation
  - Buffering
- Regional Parameters
  - Enzyme kinetics, nucleotide rise, diffusion etc.
- Well Parameters
  - Hydrogen ions generated, buffering, DNA copies etc.



Parameter Estimation

- Rich data fitting on first 20 flows
  - Custom Levenberg-Marquardt algorithm
  - Multiple parameters are estimated
  - Requires a full matrix solve like Cholesky decomposition
  - Required for each well
- A two parameter fit is required in rest of the flows
- Generates signal value corresponding to hydrogen ions generated in each flow

life technologies

Levenberg Marquardt (LM) Algorithm

- Least squares curve fitting where sum of squared residuals between observed and predicted data is minimized

where $$S(b) = \sum_{i=1}^{m} [y_i - f(x_i, b)]^2$$

S: sum of squared residuals between observed and predicted data

β: set of model parameters

y, x: observed data

- Essentially Gauss Newton (GN) algorithm with a damping factor λ tuned in every solve iteration to progress in the direction of gradient descent

LM algorithm cont'd

- For our application
  - Minimize the residual between raw data and signal value obtained from the mathematical model for each well
  - Provides numerical solution for the parameters governing the model
  - Iterative algorithm executed repeatedly till there is no appreciable change in parameters
  - Convergence in reasonable time and iterations depends on the initial guess for parameters

life technologies

GPU Acceleration

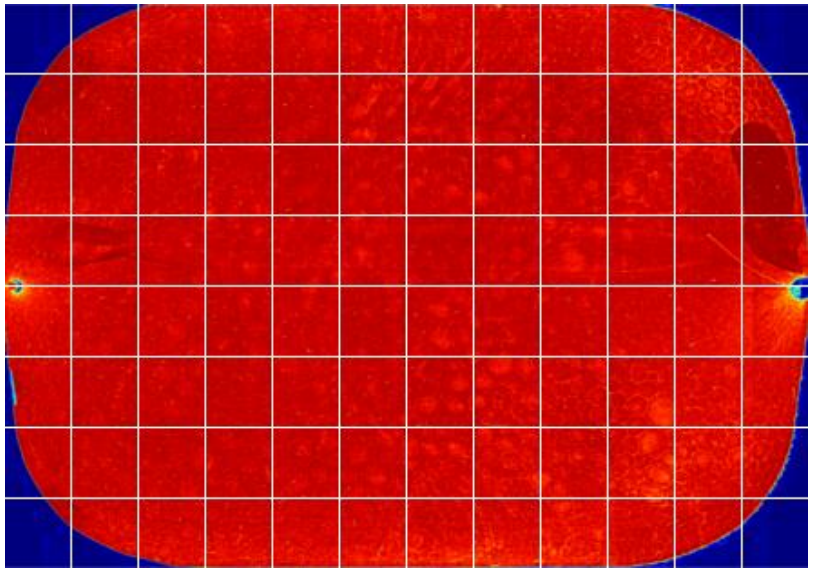
GPU Acceleration

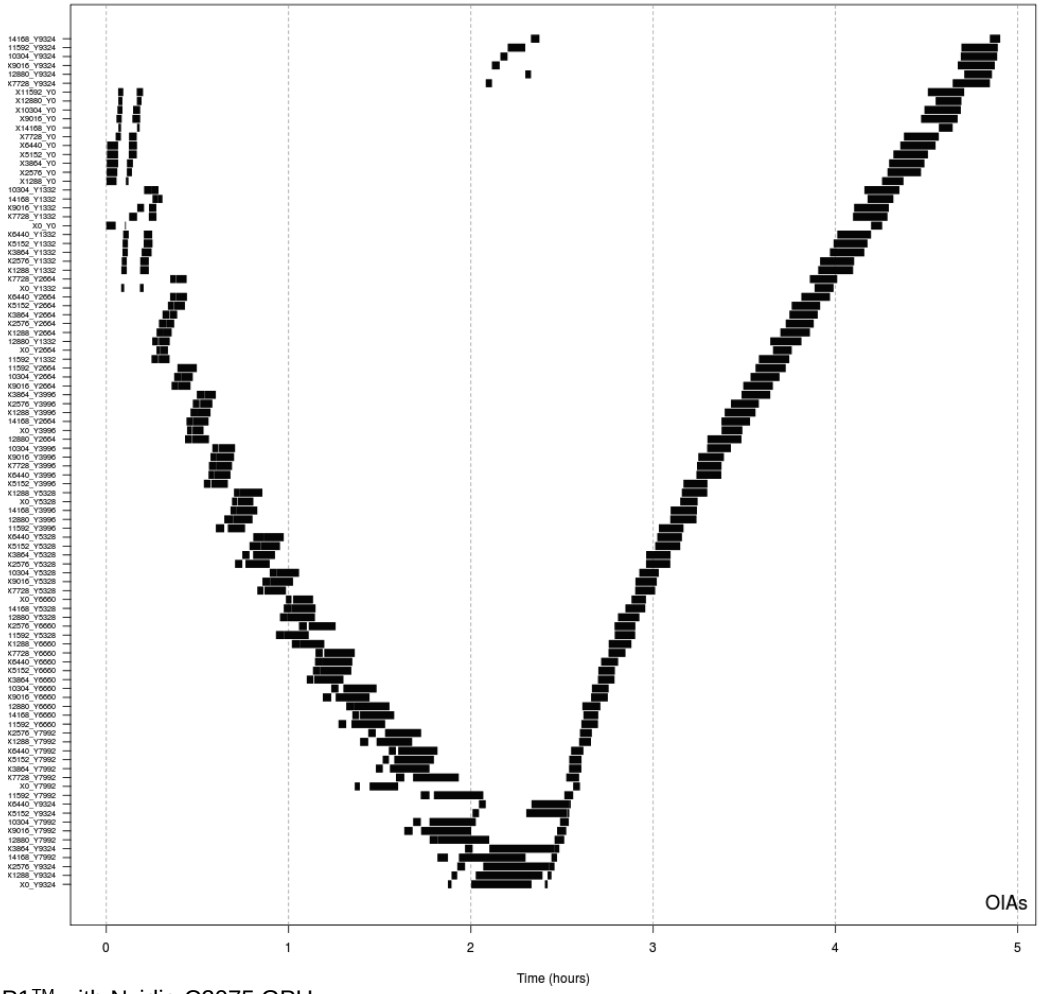
life technologies

ThermoFisher SCIENTIFIC

Current Execution Model

- Based on Original CPU implementation: Process Level



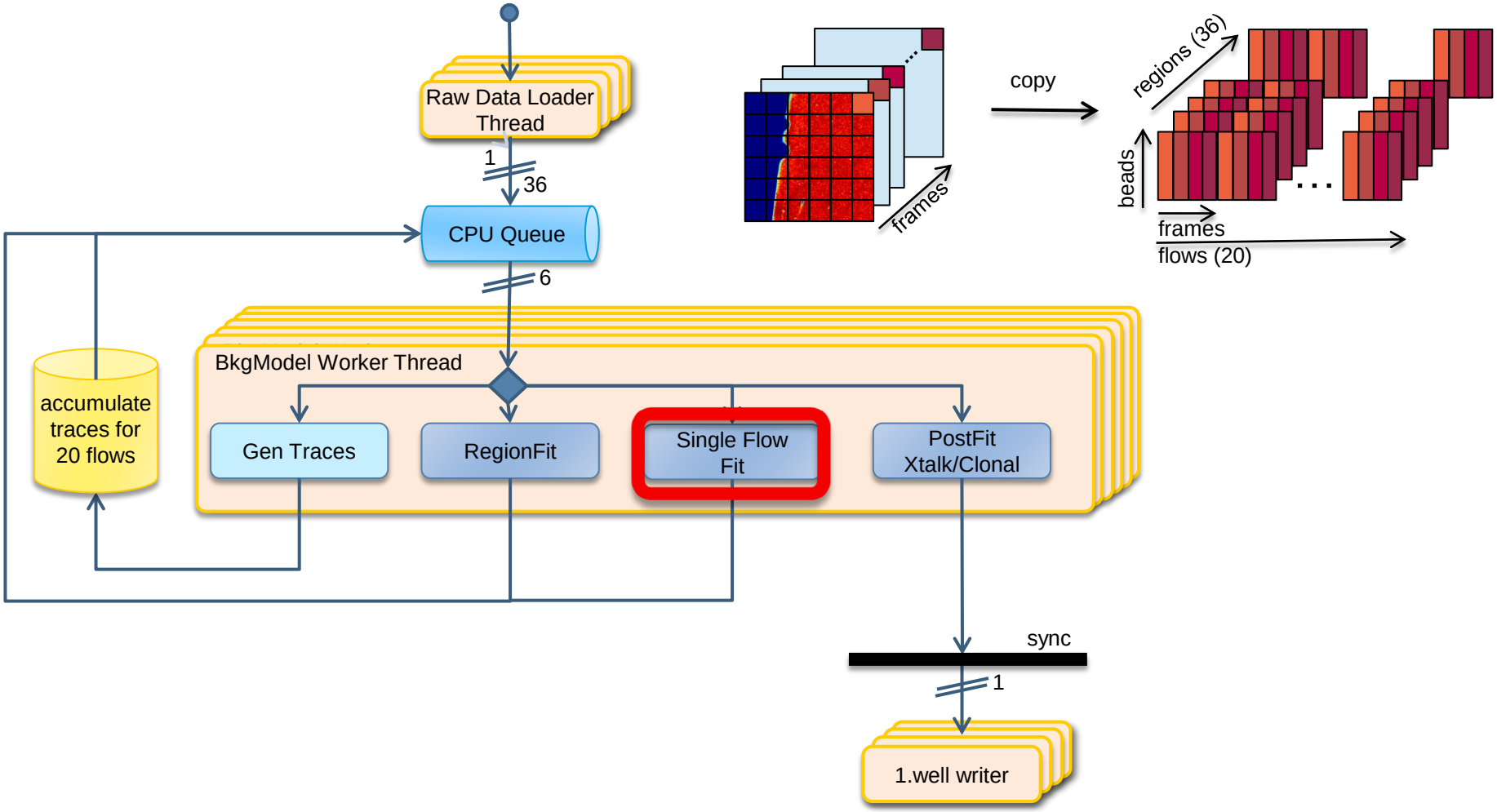


- 96 blocks
- depending on hardware 4 to 6 processes in parallel
- work on available data during experiment

*Heat-map and timing from a Proton P1™ with Nvidia C2075 GPU

Current Execution Model

- Based on Original CPU implementation: Thread Level



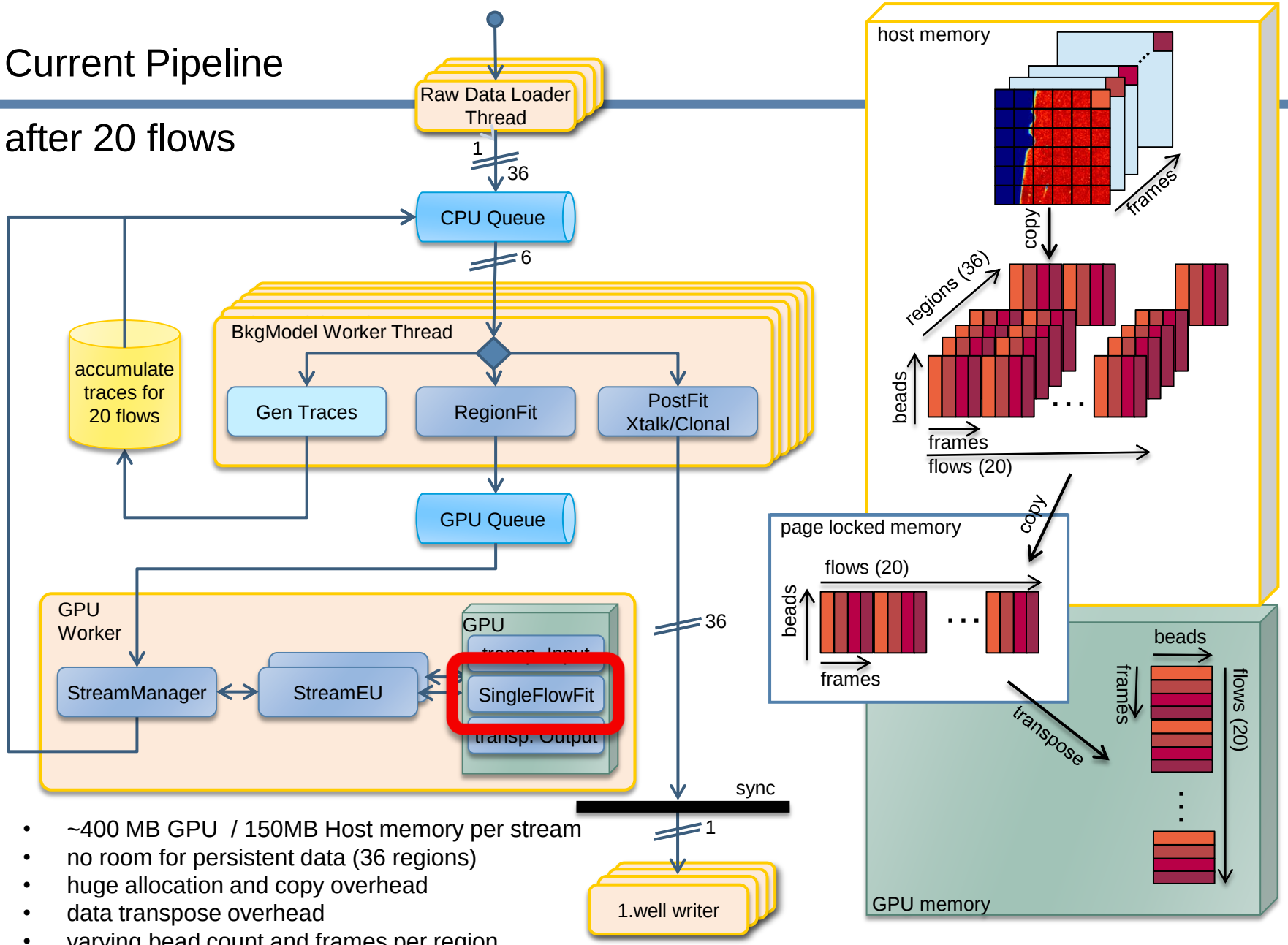
Current Implementation

- Stream based to hide PCIe transfer
  - Resources needed for stream execution are pre-allocated and obtained from a resource pool.
  - If resources to create a Stream Execution Unit (SEU) are available the Stream Manager will try to poll a new job from a job queue.
  - The Stream Manager can drive multiple SEUs which can be of different types.
  - Theoretically up to 16 SEUs can be spawned in one Stream Manager if enough resources are available

life technologies

Current Pipeline

after 20 flows



- ~400 MB GPU / 150MB Host memory per stream
- no room for persistent data (36 regions)
- huge allocation and copy overhead
- data transpose overhead
- varying bead count and frames per region, reallocation and slowdown in absolute worst case.
- synchronization steps

GPU Code Optimizations

- Merging smaller kernels into fat kernels
  - No kernel launch overheads
  - Removes lot of redundant global memory reads and writes
- Invariant code motion
- Instruction reordering to allow for better caching
- Loop unrolling
- Reduction in integer operation for address calculations
- Reduction in register pressure
  - Occupancy was register limited
  - Did a sweep over kernel launch bounds to arrive at optimal value

GPU Memory Optimizations

- Global memory access optimization
  - Data transform from AoS to SoA.
  - Data reorganization to allow for vec4 reads
  - Use of shared memory to store some heavily accessed elements
  - Padded buffers to allow for more efficient 128byte segment reads
  - “Excessive” use of constant memory
  - Optimization for L1 cache

  - needs to be revisited for every new architecture
    - e.g. Fermi to Kepler:
      - no more L1 for global r/w → __ldg(),
      - changes in shared memory per thread.
      - new L1/Shm config

life technologies

Why are further optimizations needed?

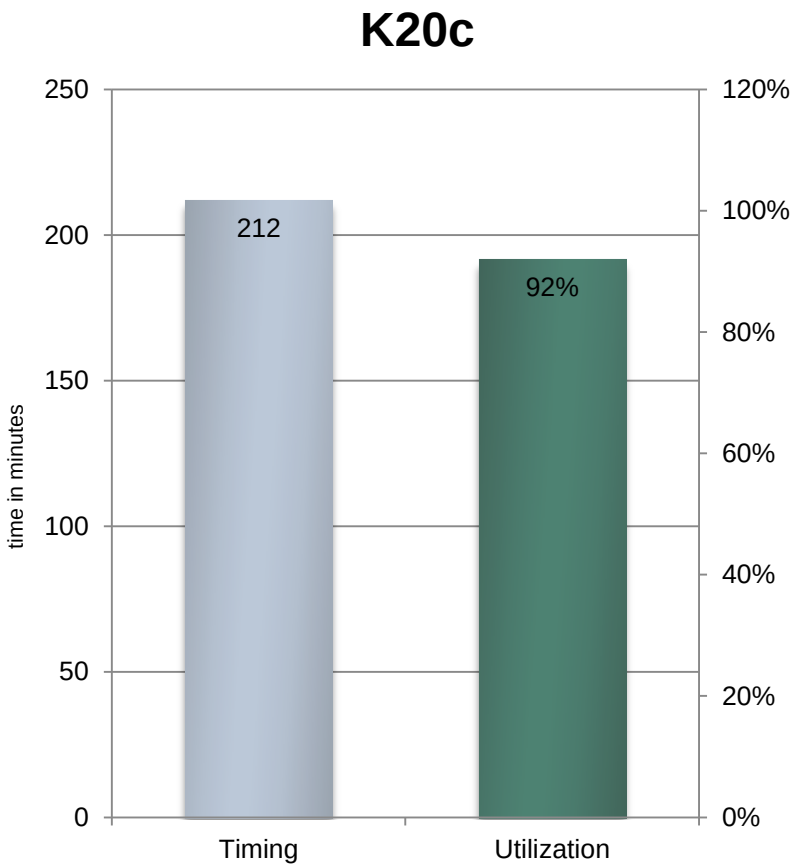
- Current pipeline optimized for PI™ signal processing
  - Utilizes GPU (more or less) efficiently during bkgmodel fitting
  - Generating empty and bead traces a bottleneck
    - Big chunk of CPU time spent in these computations
    - Mostly memory bound and a natural step to be performed on GPU as a precursor to fitting
  - Raw data processing is another big compute hog
    - This pipeline will enable it to be easily streamlined in the new flow
  - Many unnecessary data transformations and memcopies
  - Complex execution model
- All above steps will take 4x more time with PII™
  - No filtering so pure 4x increase in time

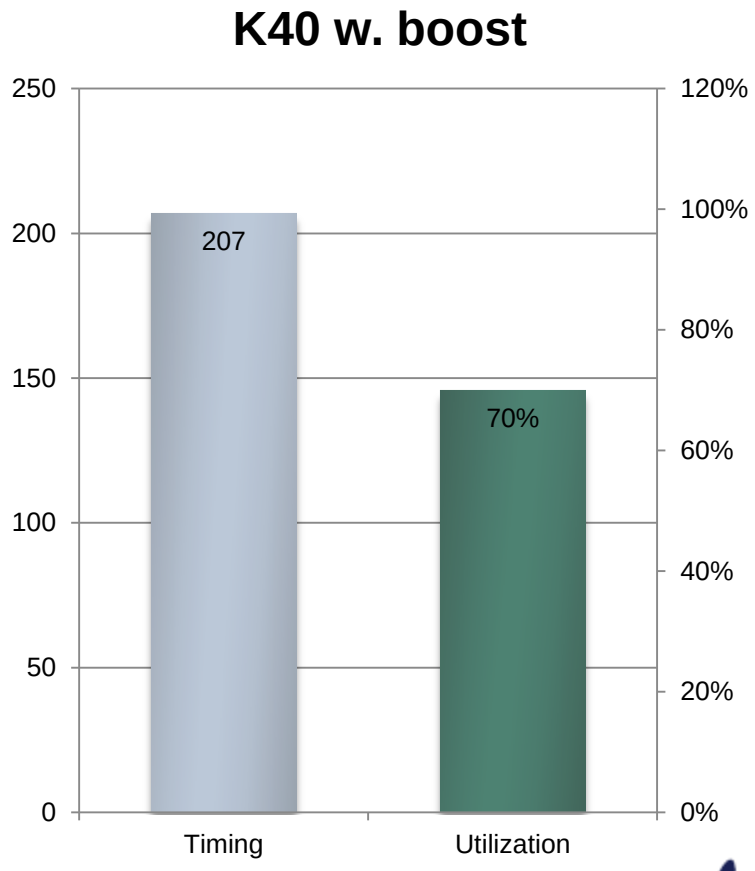
life technologies

Why are further optimizations needed?

Wall clock time and GPU utilization for K20c and K40 w. boost

(Proton P1 experiment run with 500 flows)





life technologies

Signal Processing Flow

Reading flow data

Raw Data Processing

Regional Parameter Estimation (Common to all wells)

Parameter Estimation unique to each well (LM fitting)

Post Fit Processing

Writing signal values

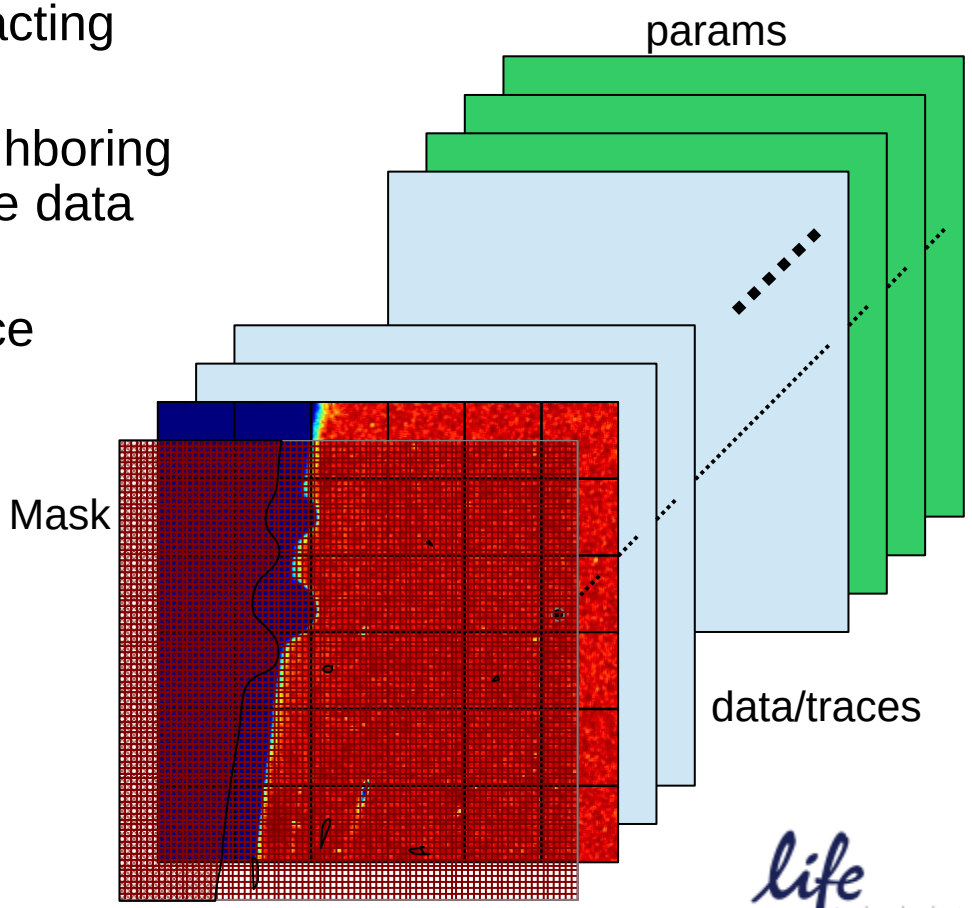
Current Optimization Work

- Expand scope of GPU implementation
- Some algorithm tweaks
- Modifications in intermediate data layout
- Removed need for addition copies and transposes
- Changes in spatial and temporal data subdivision
- Use of Nvidia MPS to hide PCIe transfers

life technologies

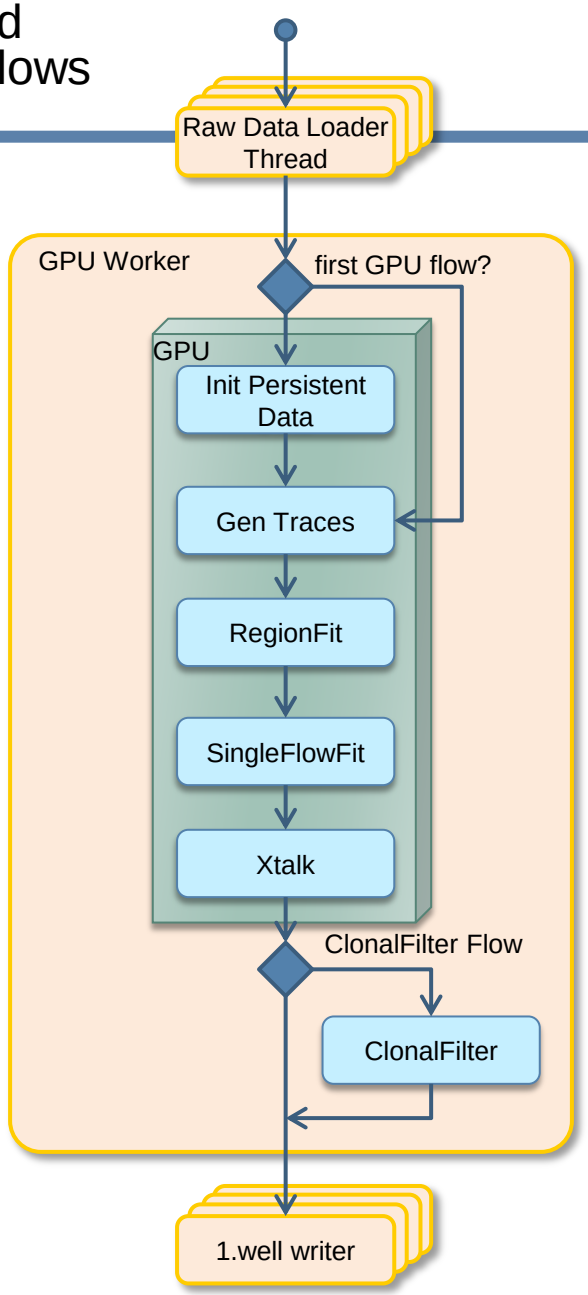
Data Layout

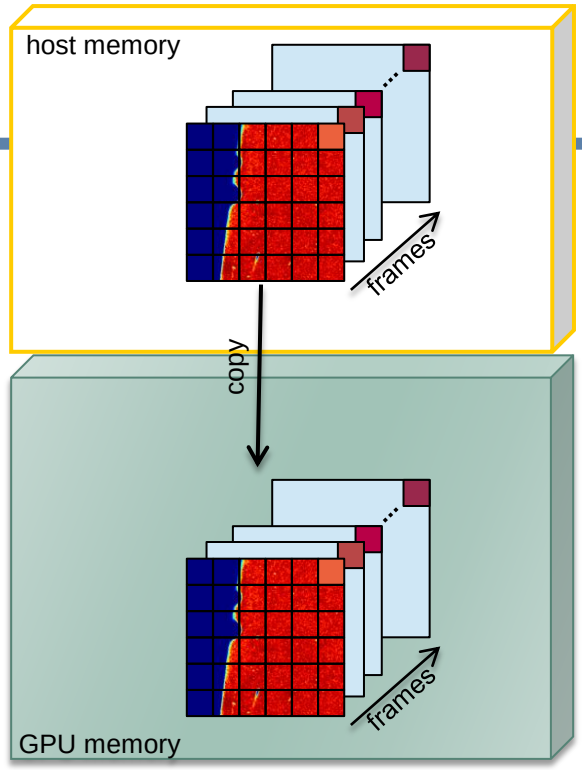
- New Layout
  - Maintain layout of raw data
  - Use mask on GPU instead of extracting relevant data from raw data
  - Keep all the data in planes so neighboring threads always access consecutive data elements
  - Fixed data layout allows for in-place recompression and other operations
  - Easier resource management
  - More data to work on for GPU
    - instead of 20x now 36x region size
  - Region size does no longer limit GPU performance



life technologies

Optimized after 20 flows



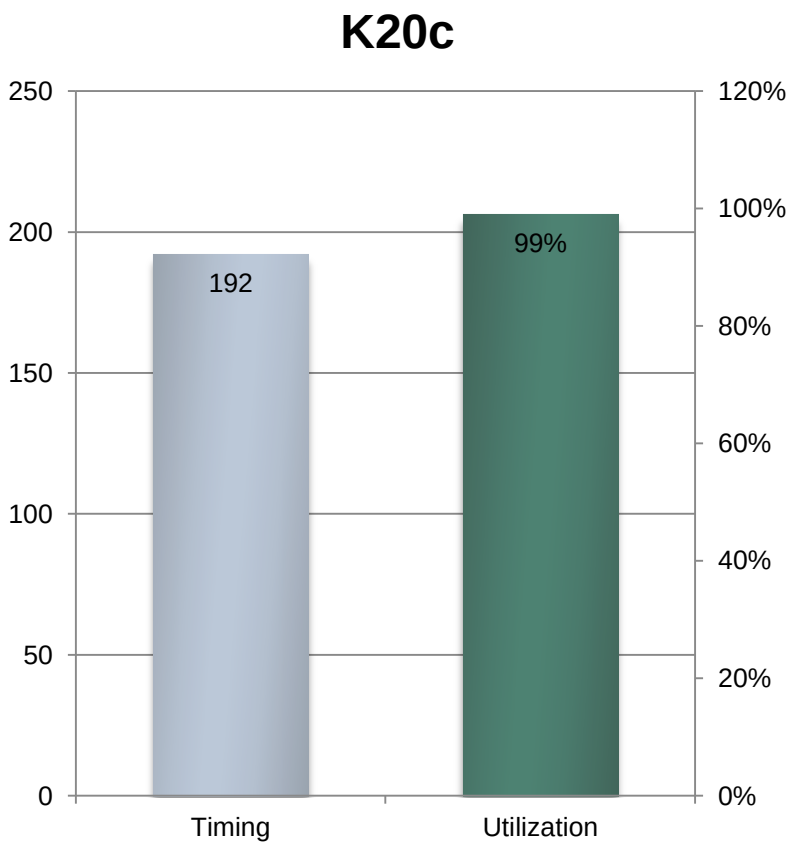


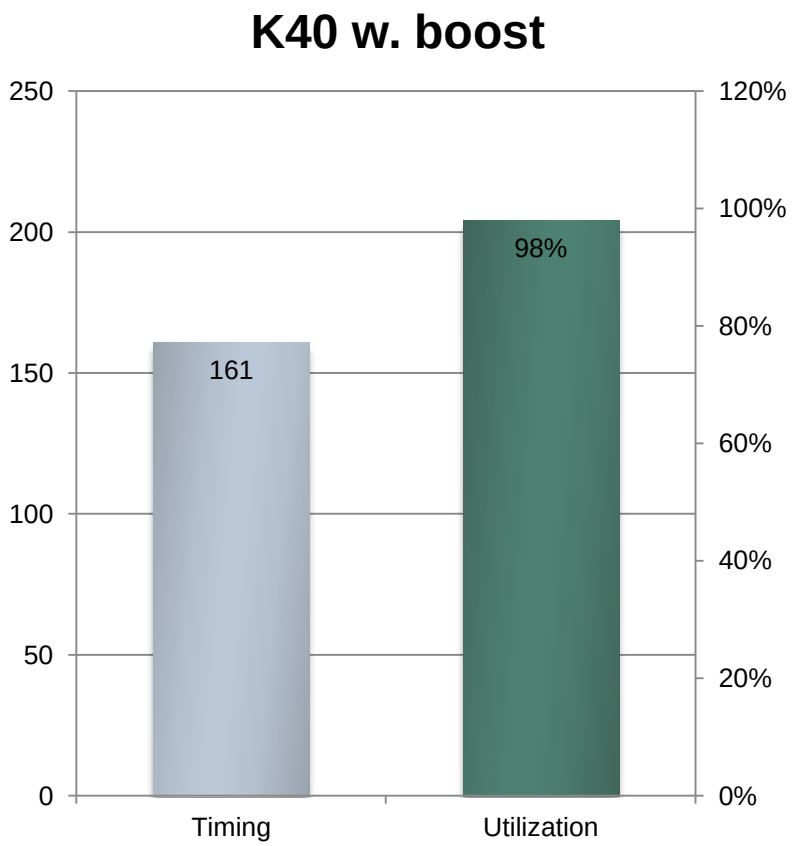
- Per block fixed amount ~270 MB GPU memory
- Almost no additional host memory
- Persistent data only copied/generated once on device, no additional transposes.
- No host side copy overhead (use of MPS to hide PCIe)
- Fixed max block size, no need for re-allocation
- No synchronization steps

optimize pipeline performance

Wall clock time and GPU utilization for K20c and K40 w. boost

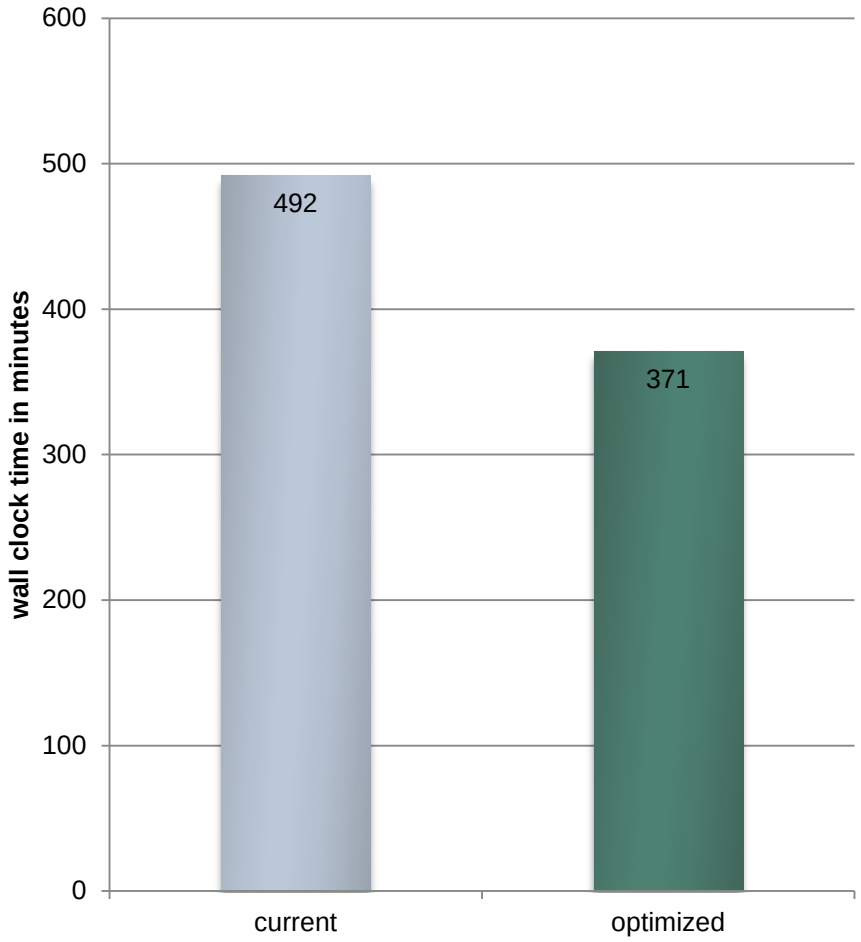
(Proton P1 experiment run with 500 flows)





life technologies

optimize pipeline performance for 300 flow PII™ run

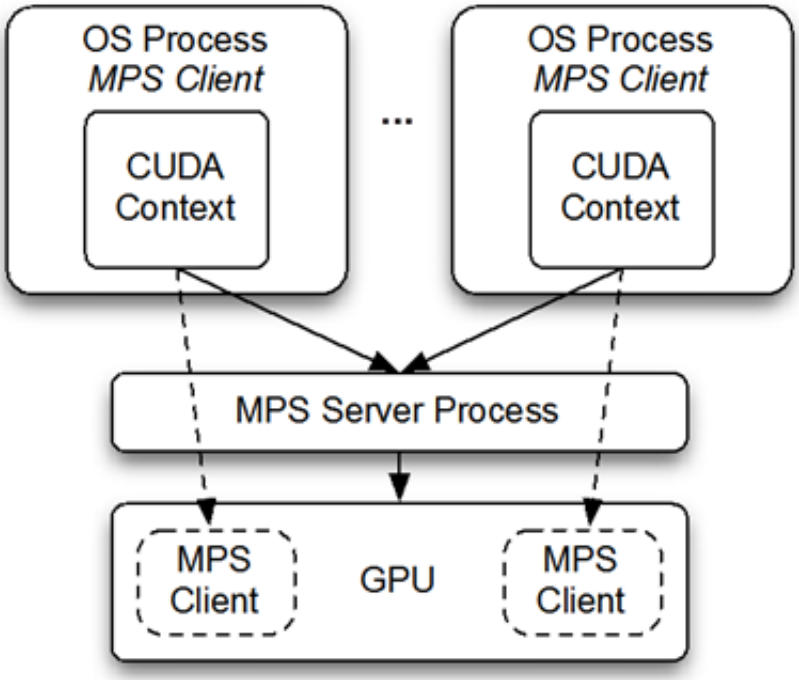


optimized pipeline 25% faster than current implementation.

life technologies

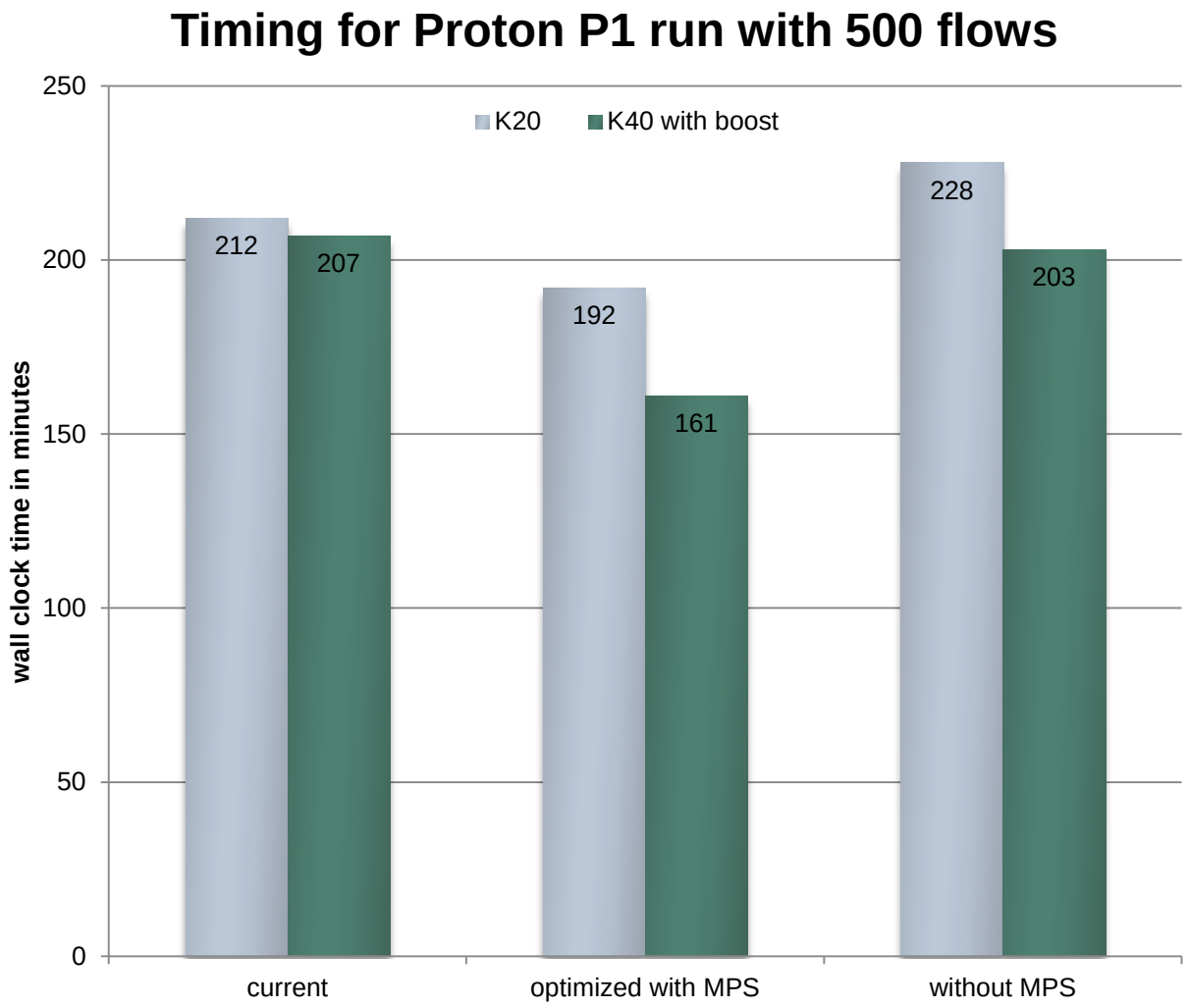
NVIDIA MPS

- Multiple overlapping contexts on one device
- concurrent PCIe copies and kernel execution
- Huge reduction in code complexity → no more streams



life technologies

MPS performance



MPS issues

- occasional slowdown (overhead?)
- undefined state after a process encountered an exception/error

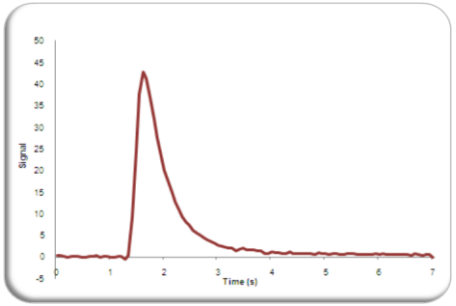
Optimization Summary

- P1™/P2™ block level signal processing
  - Concept of bkgmodel regions internal to the GPU
  - Easy to experiment with different region sizes
  - Regions can talk to each other.
- Streamlined flow from raw data processing to signal processing
  - Sequential execution of the pipeline steps
  - Final output to be written to 1.wells
  - Fewer data copies, reduced memory footprint for P1™
  - Freed up CPU resources
- Reduced context switches on the GPU
- Better utilization of PCIe bandwidth

life technologies

Expanding the Scope

21 TB
(PII™ chip)



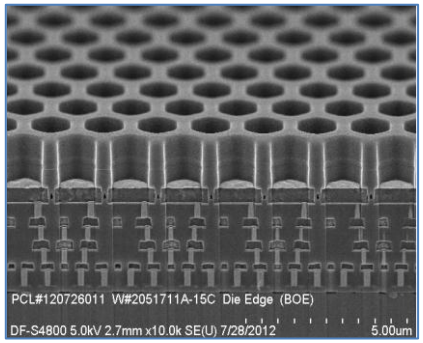
Data acquisition and compression in FPGA

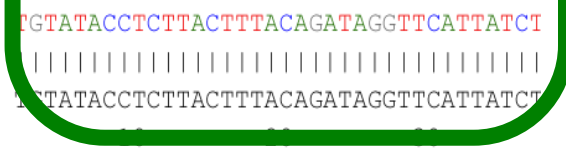
2 TB

Signal Processing

200 GB

BaseCalling and Alignment to reference genome





GPU-based Mapping

- Aligners considered
  - **TMAP**
    - Mapper from Ion Torrent developed in-house
    - Dual 8-core Intel® Xeon® Sandy Bridge
  - **nvBowtie**
    - Re-engineered version of bowtie2 from NVIDIA
    - NVIDIA® Tesla® K40
- Dataset
  - Ampliseq Exome from Ion Torrent

life technologies

nvBowtie: Speed and Quality

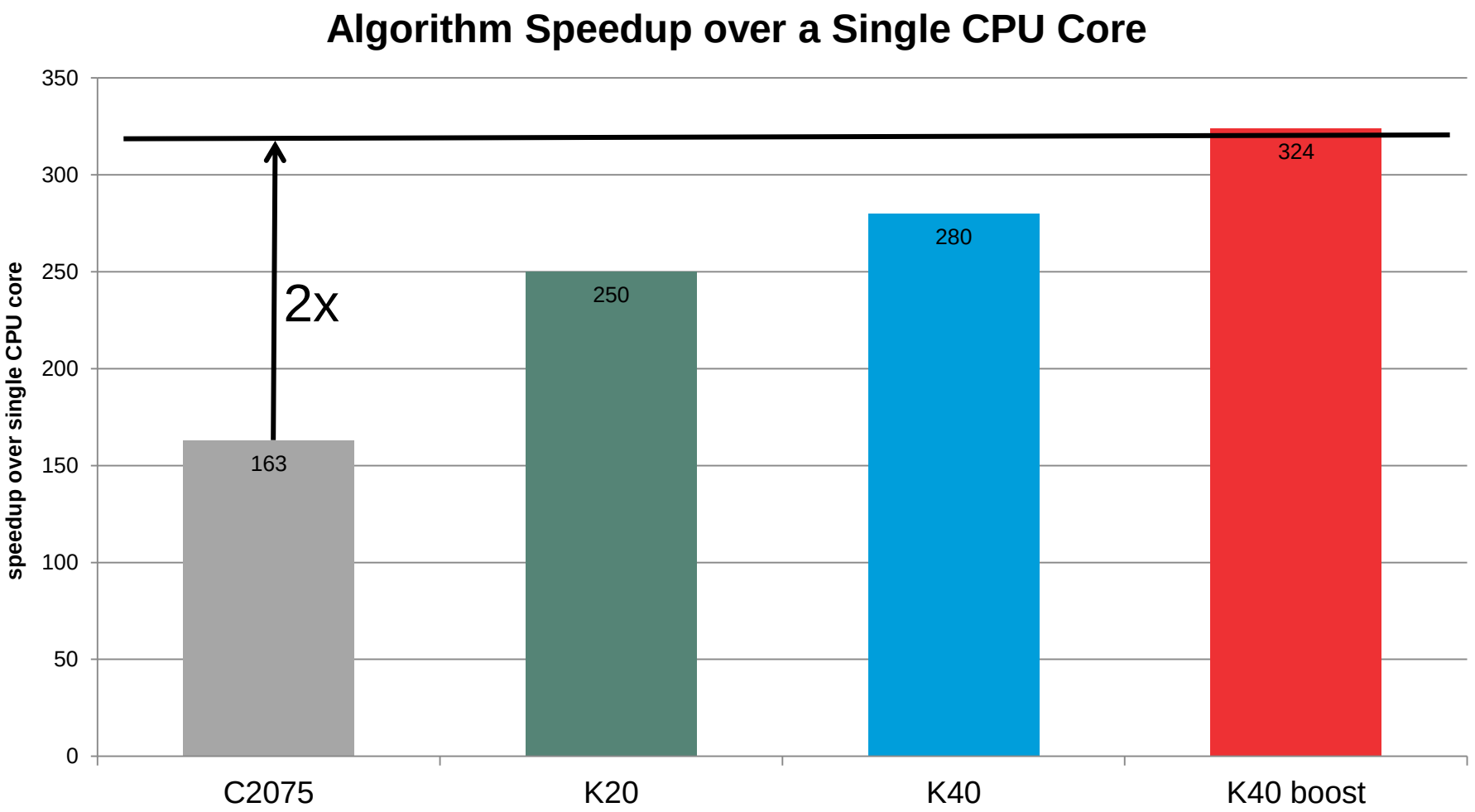
- Speed
  - Index construction (using H.sapiens genome assembly v.39)
    - **14X** speedup
  - Mapping (45M reads)
    - **7.1X** speedup
- Quality
  - Mapping Concordance
    - 100% on unique locations
    - 51% overall (**Seems to be a bug with the most recent version of nvBowtie. Actively investigating with NVIDIA)**

Thanks to Denis Kaznadzey for evaluating nvBowtie and generating timing and performance results

life technologies

Conclusion

Comparison GPU generations



Conclusion and Future Work

- Breach the legacy code wall.
  - CUDA quickly speeds up bottlenecks until your legacy code prevents you from scaling. At that point severe refactoring might be the only solution.
- Expand the scope
  - Other pipeline components need more attention, e.g. Alignment
    - Well known algorithm Smith waterman for detailed alignments
    - NVBIO from NVIDIA is the preferred solution
- More work to do to tackle the PII™ data challenge
  - Exploit Kepler architecture to its maximum potential
  - Explore potential of Maxwell GPUs. preliminary test are promising

life technologies

Thank You

NVIDIA

specifically the DevTech Team

Nikolai Sakhkarnykh

Jonathan Bentz

Jacopo Pantaleoni

Mark Berger

Kimberley Powell

Our supervisor Charles Sugnet, Eugene Ingermen

and

The entire Ion Torrent R&D team

life technologies

For Research Use Only. Not for use in diagnostic procedures.

life technologies

GPU Acceleration

Appendix

life technologies

LM algorithm

- Need to solve the following equation for δ

$$\left(J^T J + \lambda\, diag(J^T J)\right)\delta = J^T\left(y - f(\beta)\right)$$

where

J: Jacobian matrix

δ: Increment vector to be added to parameter vector β

y: vector of observed values at (x_0 , . . . , x_n)

f: vector of function values calculated from the model for given vector of x and parameter vector β

λ: damping parameter to steer the movement in the direction of decreasing gradient

LM algorithm cont'd

- Run an outer loop on some maximum number of iterations
  - Exit if no appreciable change in parameter value from previous to current iteration
- Run an inner loop
  - Worse case loop count depends on max or min value of λ
  - Increase λ if residuals increase compared to previous outer iteration
  - Decrease λ if residual decrease compared to previous outer iteration

life technologies

LM implementation on GPU

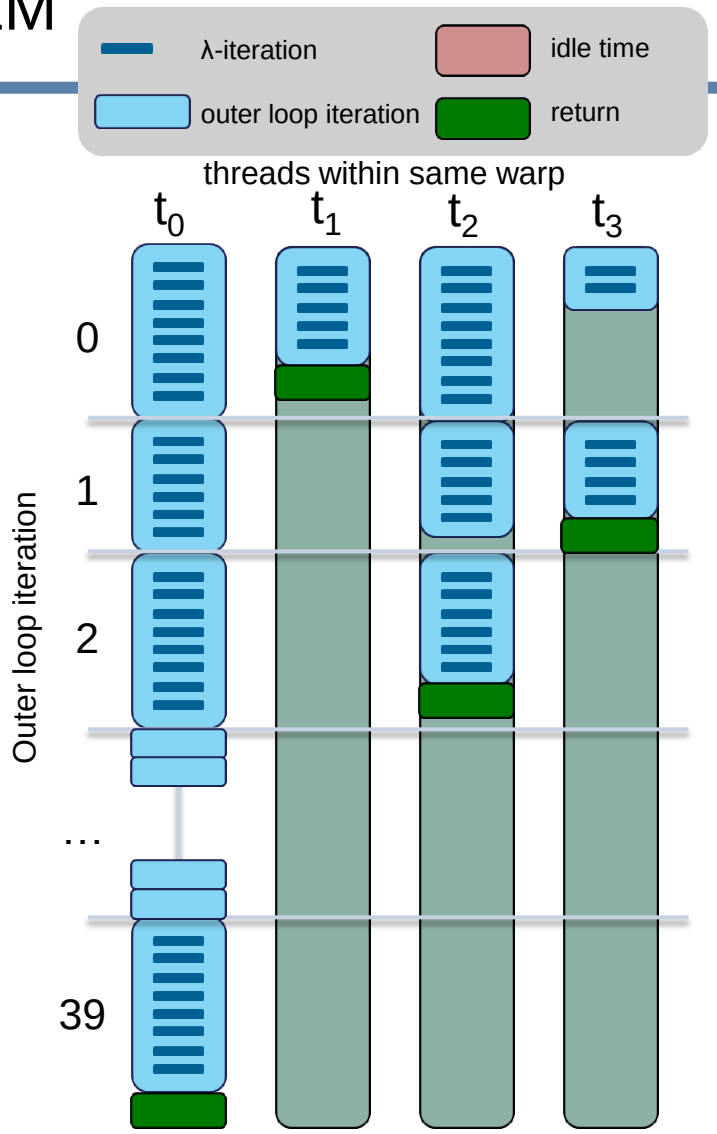
- LM for each well is done by one thread on GPU
- Millions of wells
  - Embarrassingly parallel problem
  - High on compute
  - Several iterations required to obtain a solution with a reasonable tolerance
- Thread exits once numerical solution is obtained for the parameters

life technologies

Bottlenecks of GPU implementation of LM

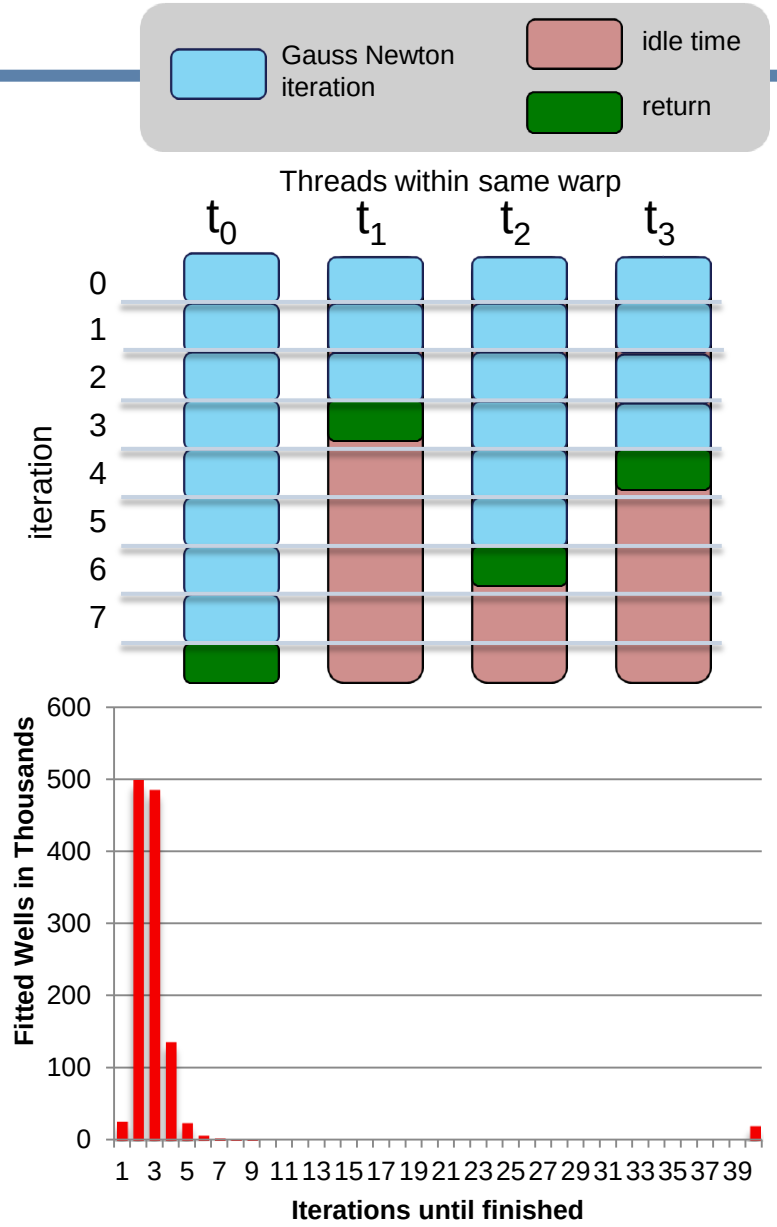
Divergence among GPU threads in a warp

- Each well is sequencing different DNA fragment
- Need different number of iterations
  - Different number of iterations of increasing or decreasing λ for each iteration
- Towards the end of the algorithm execution, only few threads are active in a warp
- Scarce resources on GPU doesn't allow LM kernels from two different jobs to overlap



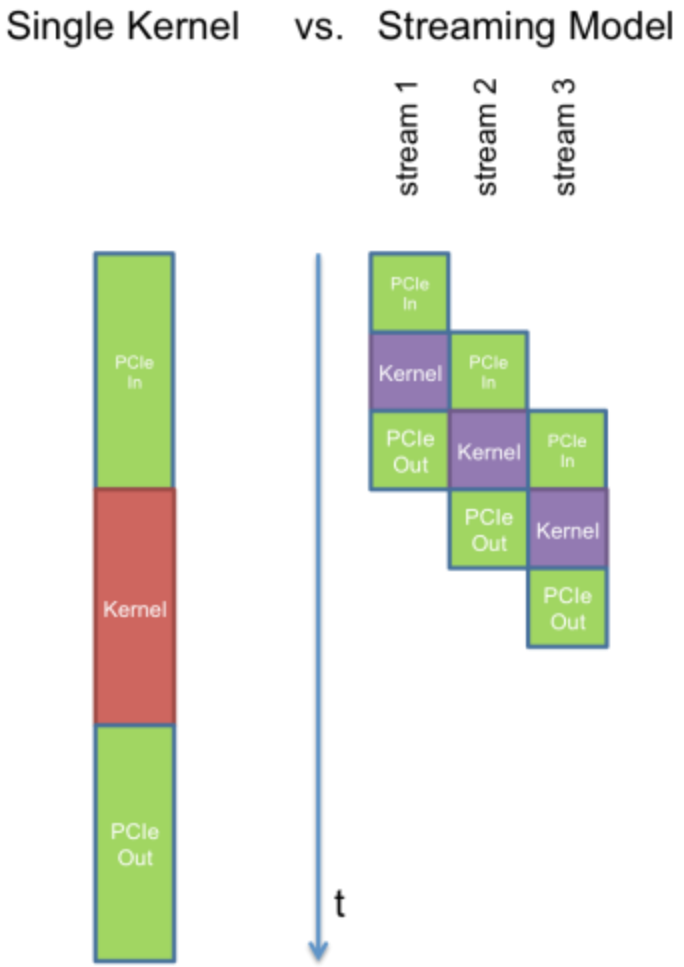
Switch to Gauss Newton Algorithm

- Same as LM algorithm without damping parameter λ to be tuned in every iteration
- Converges faster if initial guess is in the vicinity of the solution
- Mitigated the divergence problem caused by LM to a great extent
  - No λ tuning iterations are involved
- Reduced code and complexity
- Limit maximum iterations to be performed in worst case scenario



Execution Model

- Stream based execution allows for overlap of
  - Kernel execution
  - PCIe transfers
  - host side data reorganization
  - CPU pre and post processing
- Queue system for heterogeneous execution
  - Balances execution of different tasks between CPU and GPU



life technologies