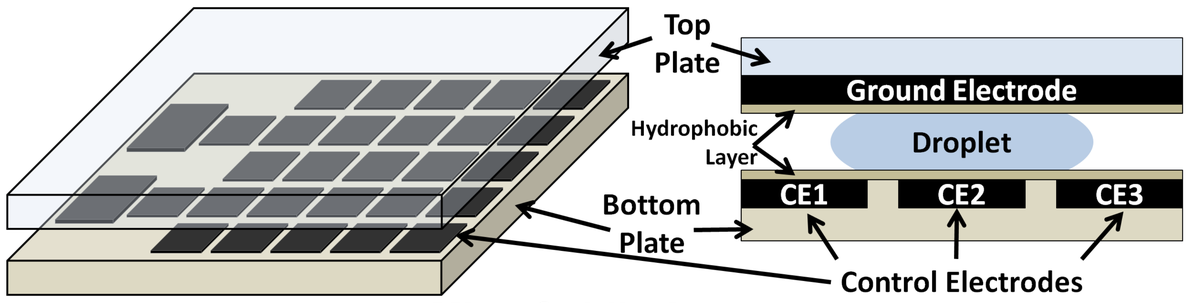
Path Scheduling on Digital Microfluidic Biochips

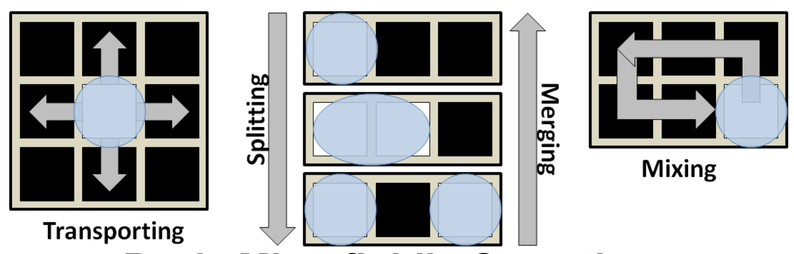
Dan Grissom and Philip Brisk
University of California, Riverside (UCR)

Digital Microfluidic Technology

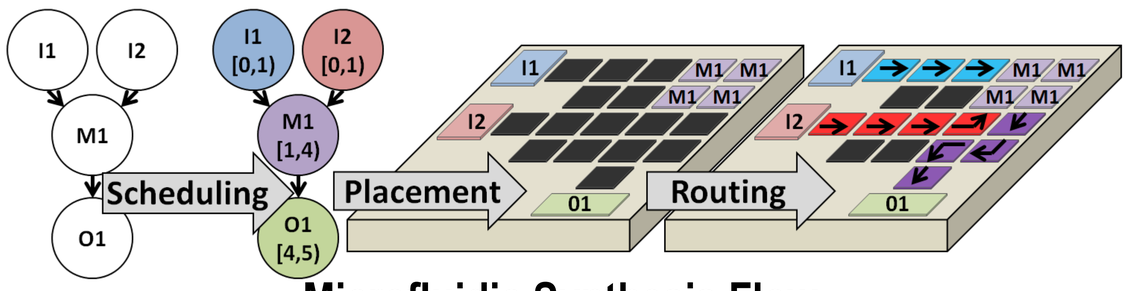
Digital Microfluidic Biochips (DMFBs) are an emerging "lab-on-a-chip (LoC)" technology that perform biochemical reactions by operating on fluidic droplets on the scale of nano-liters.



Microfluidic Array



Basic Microfluidic Operations



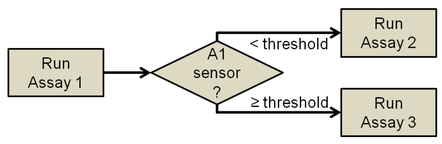
Microfluidic Synthesis Flow

High-Level Motivation and Problem

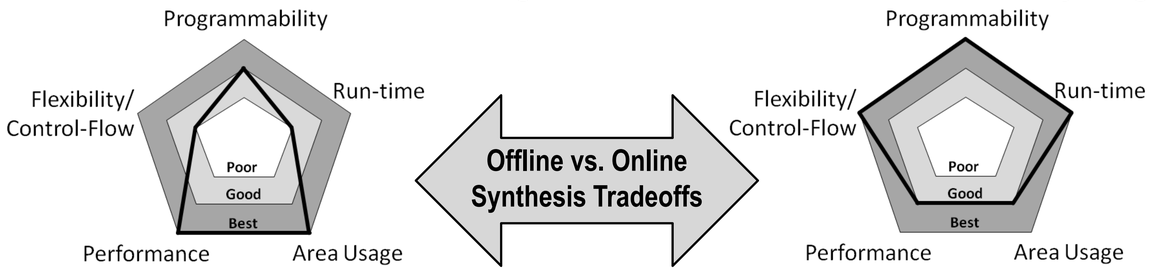
Goal: Online synthesis

Motivation: To enable DMFB programmability and new features in the areas of control-flow and live-feedback

Example: A control-flow graph which can dynamically decide which assay to run next based on live feed-back from the DMFB:



Problem: Past offline synthesis methods are computationally complex, which would add significant amounts of time to the assay length



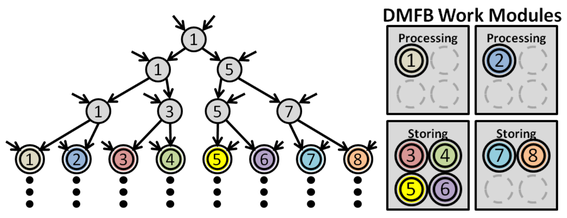
Solution: Develop synthesis methods that yield good results in little time so the overall assay length is kept short

Minimizing Storage

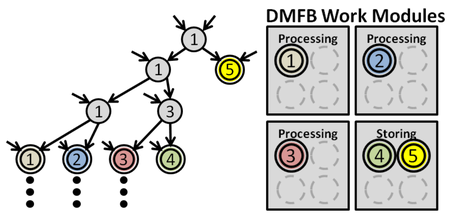
- Traditional computing has "infinite" storage, as far as program is concerned

- DMFBs have limited storage since the same cells used for operations must also be used for storage if droplets are not ready to be operated on

- The order in which we schedule nodes can affect the amount of useful work the DMFB is performing:



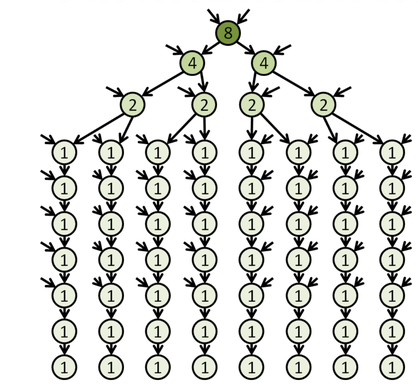
List Scheduling w/ priority function favoring nodes w/ longer paths to output



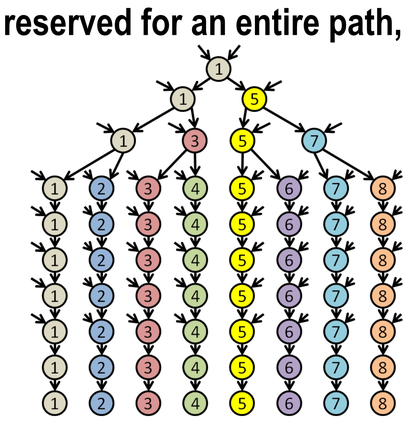
Path Scheduling w/ priority function favoring paths with less fan-out

Path Scheduling Algorithm

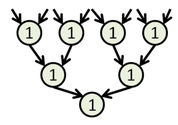
- Compute schedule for an entire path at a time (instead of node at a time)
- Start with a path-leader
  - Initially a node with only dispense parents
  - If there is a branch on the path:
    - Reserve resources for the node with the lowest IPP; continue down path
    - All other branch nodes are added to the list of path leaders for later
- Once resources have been reserved for an entire path, the path is scheduled



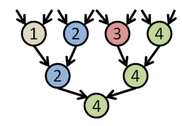
Protein Independent Path Priorities (IPPs)



Protein Paths



PCR Mix IPPs

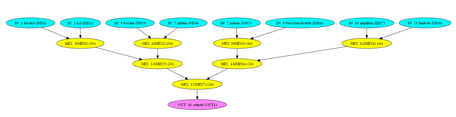


PCR Mix Paths

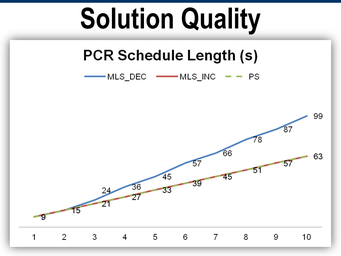
Results

PCR Mixing Stage Assay

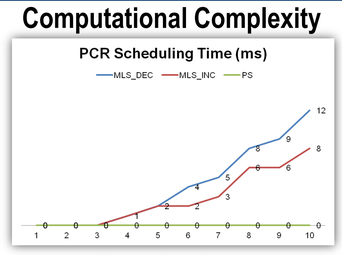
- Equivalent results in slightly less time



Solution Quality

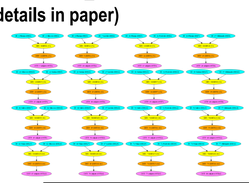


Computational Complexity

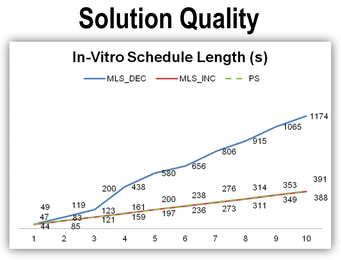


In-vitro Assay

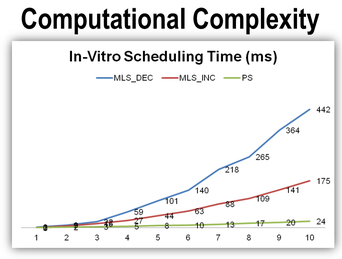
- PS 2-3 time-steps slower than MLS_INC
  - Input inefficiency (details in paper)
- PS 3x to 7x faster



Solution Quality

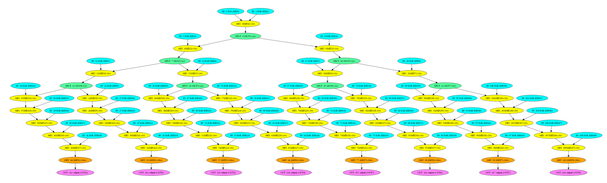


Computational Complexity

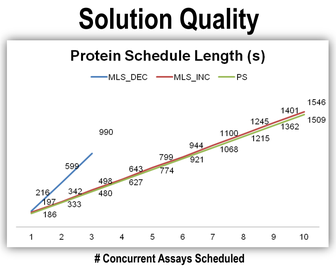


Protein Assay

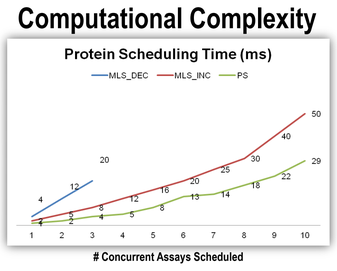
- MLS_DEC cannot schedule many DAGs b/c of storage inefficiency
- PS chops dozens of time-steps off schedule
- PS runs in about ½ the time of MLS_INC
- PS uses the fewest modules for storage
- As number of modules increases, solution quality converges



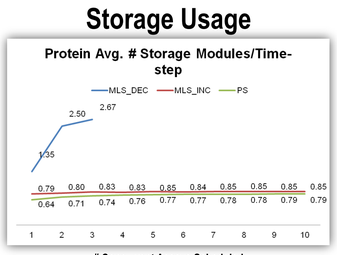
Solution Quality



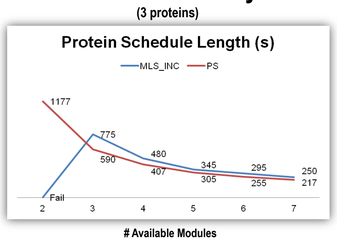
Computational Complexity



Storage Usage

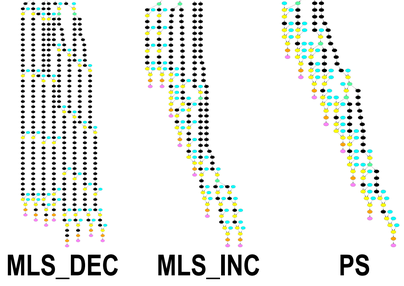


Solution Quality
(3 proteins)



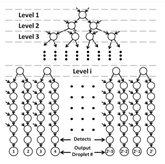
Scheduled/Placed Protein:

(Black node = droplet being stored for a # of time-steps in 1 module)

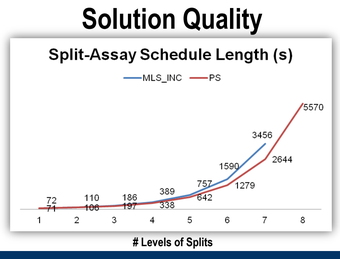


Split-Level Protein Assay

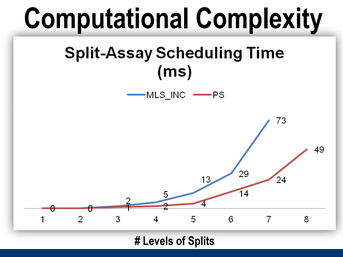
- PS saves hundreds of seconds as fan-out increases
- PS saves several dozen milliseconds of computation time
- MLS Fails at 8 splits



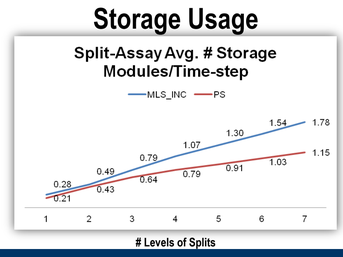
Solution Quality



Computational Complexity



Storage Usage



Contact

Dan Grissom (grissomd@cs.ucr.edu)
Philip Brisk (philip@cs.ucr.edu)
Computer Science Department, Bourns College of Engineering
University of California, Riverside