Optimizing cluster-robust variance estimation on GPUs

Scott McNeil
School of Computer Science
Carleton University, Ottawa, Canada
scott.mcneil@carleton.ca

What is CRVE?

Really, really short answer:

Corrects inference when observations have correlation across groups

(school classrooms, for example)

Why is it expensive?

By itself, it's not expensive, but it's often repeated:

- Bootstrap resampling → **100-1,000X**
- Monte Carlo simulations → **10,000+X**
- Or both

The Sandwich Matrix

This is the formula for CRVE:

$$(X'X)^{-1}\sum_{g=1}^{G} x_g' u_g u_g' x_g \,(X'X)^{-1}$$

Key problem is, for each group, we calculate:

$$\begin{bmatrix} x_{11} & \cdots & x_{m1} \\ \vdots & \ddots & \vdots \\ x_{1k} & \cdots & x_{mk} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix} \begin{bmatrix} x_{11} & \cdots & x_{1k} \\ x_{21} & \cdots & x_{2k} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mk} \end{bmatrix}$$

A few assumptions: k is small, m*G is large

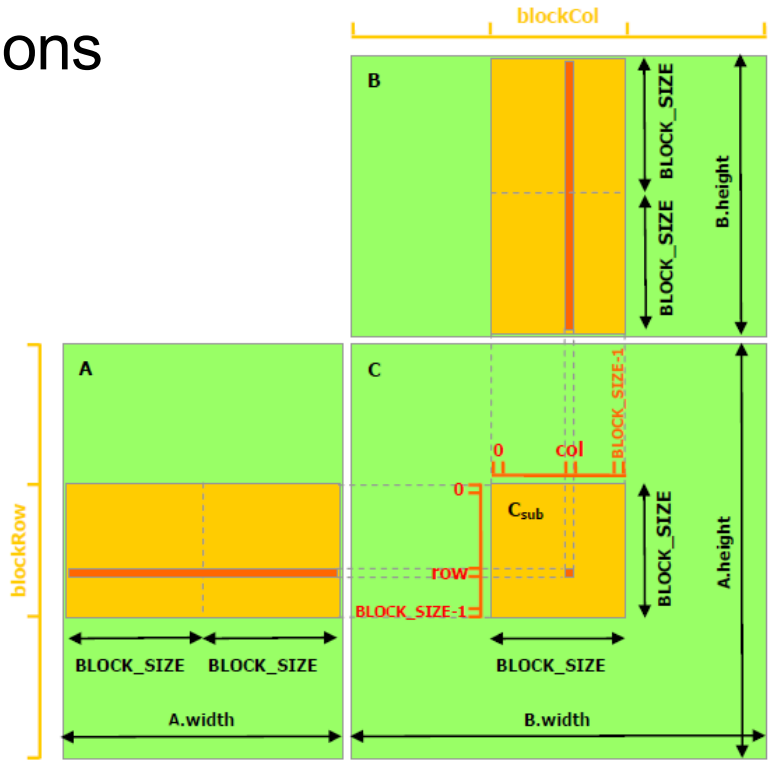
Batched Matrix Operations on GPU

Classic GPU matrix-multiplication doesn't work well when matrices are small

Libraries like Magma and cuBLAS have batched methods for many small matrix operations

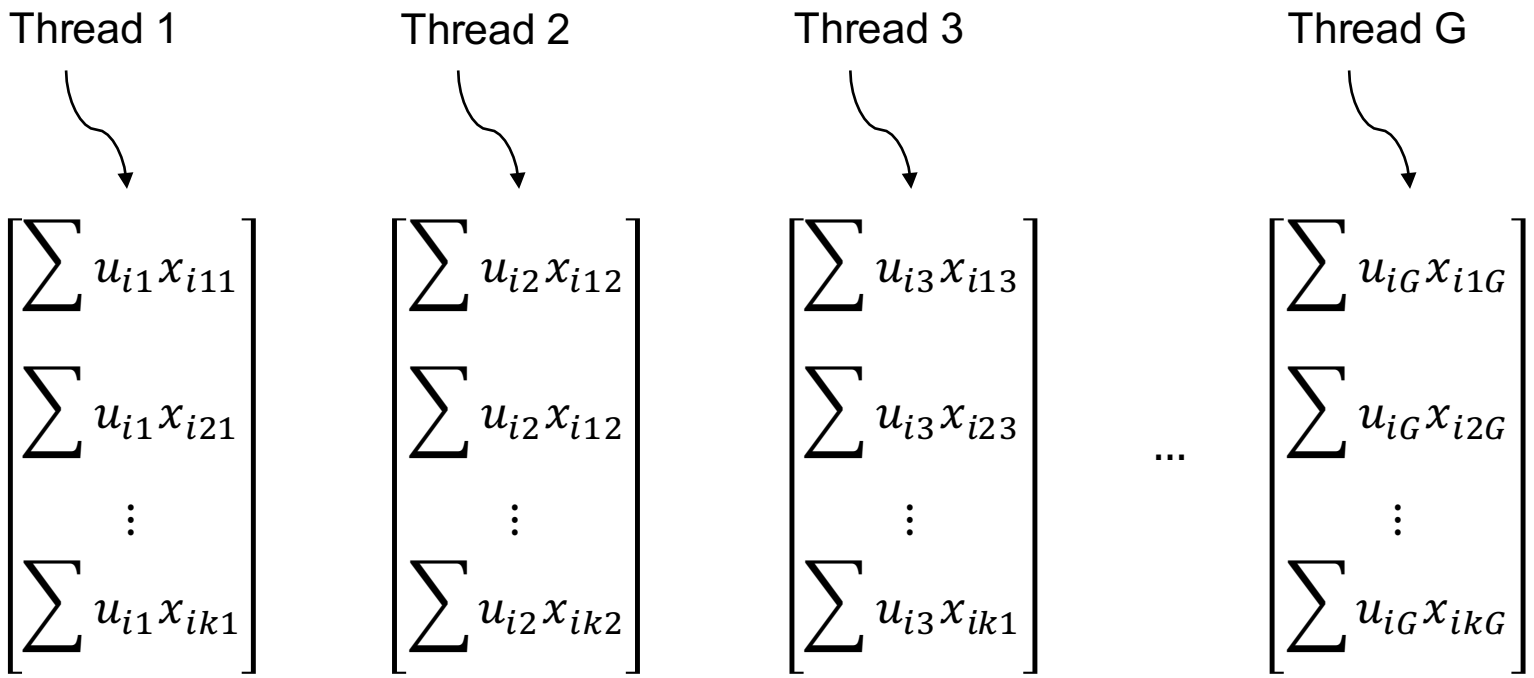
Key takeaways from literature:

- Maximize register use
- Minimize loading from memory



Inner Product

Approach: since k is small, calculate entire inner product for one group in a single thread

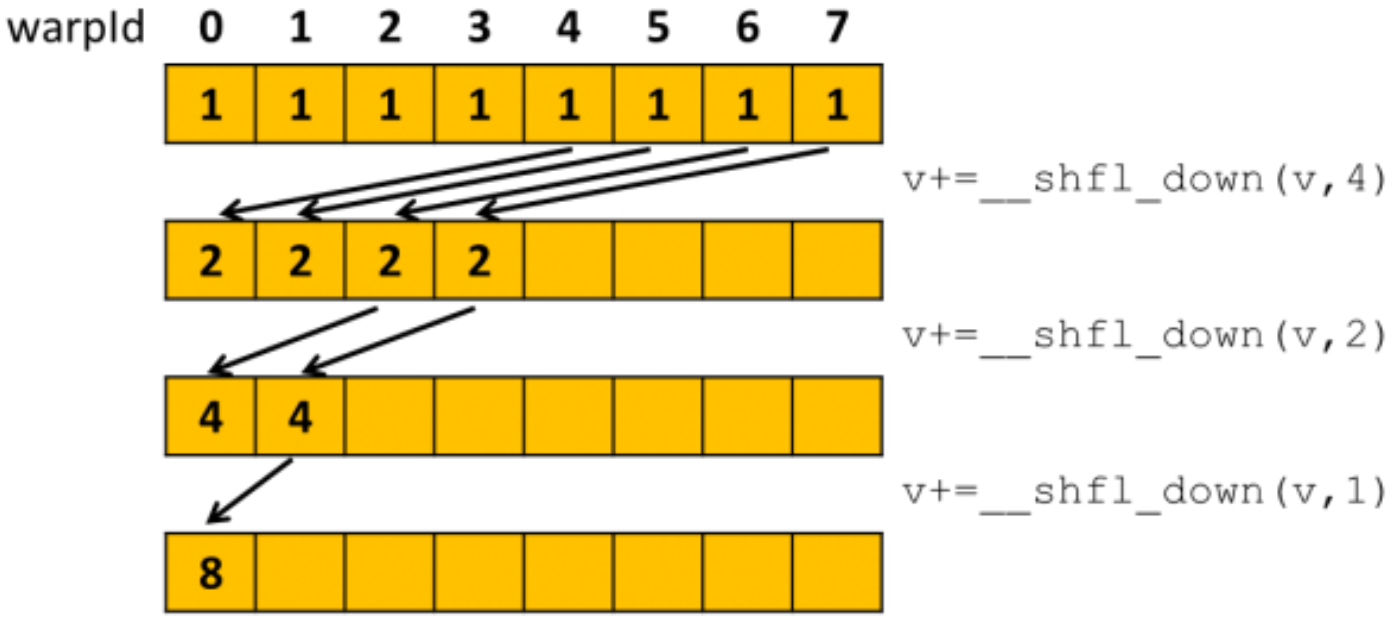


Only trick: registers aren't dynamically indexable, so need to use loop unrolling and/or meta-programming

Outer Product and Summation

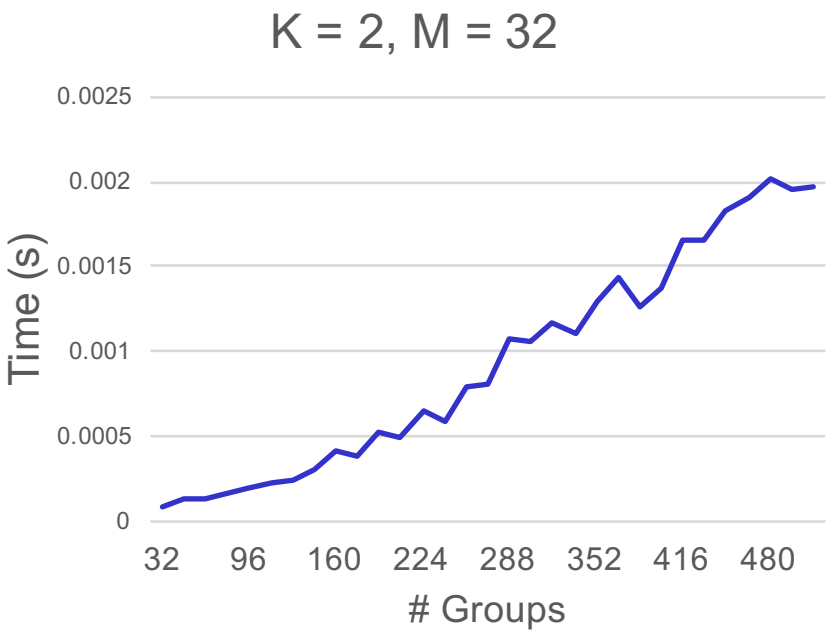
Can re-use inner product result to calculate outer product (and it's symmetric, so I just need the upper triangle)

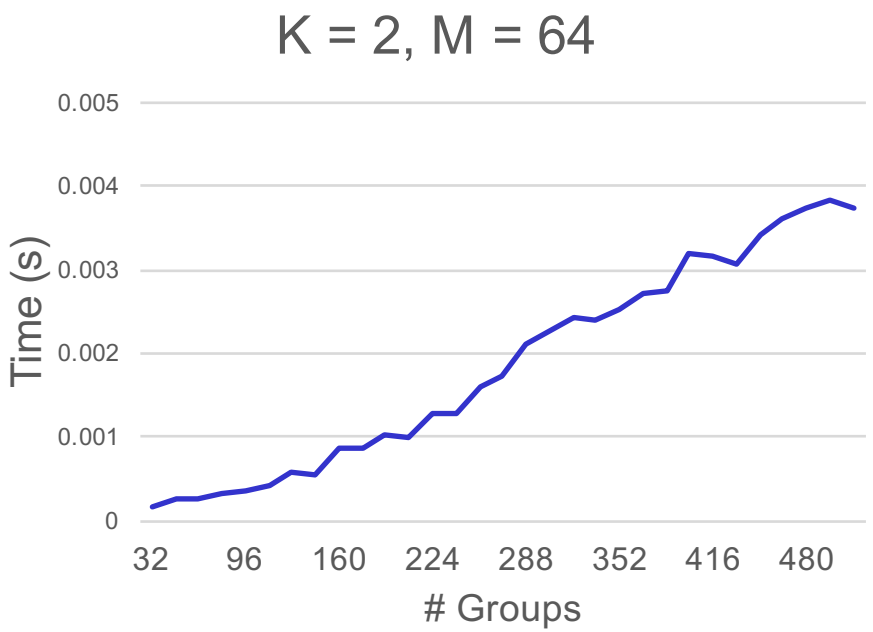
Then, since all the data is sitting in registers, I can sum the matrices via a "warp shuffle" reduction:



Very Preliminary Results

Just looking at K = 2, M = 32 and 64, 64 repetitions each





Really rough comparison: this is about a 10x speedup from sequential C++

Next Steps

Things to look at next:

- Parameter tuning – literature says this is important
- Making sure memory coalescing is correct
- Potentially combine with other steps to increase work-to-memory access ratio

Thank you!