LabVIEW Real Time for High Performance Control Applications

Aljosa Vrancic, Lothar Wenzel, LabVIEW R&D, National Instruments

Case studies: Matrix-vector Multiplication & PDE Solver

Most of the basic computational algorithms for high-performance computing are designed to have the best average performance for the most general case as they are used in off-line calculations: the goal is for calculation to end in as little time as possible for an arbitrary set of inputs but each step in general does not have a strict time deadline. As a result, they do not scale well into hard real-time embedded environments that have much stricter per-iteration timing constraints, often in a 1 ms range. We found the following approach to work much better for real-time environments:

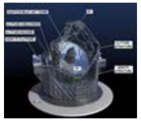
Divide algorithm into two steps:

1) Off-line calculation – it is acceptable for this step to be expensive, as it is done up front

2) On-line calculation – this step uses input and off-line calculation data to compute outputs deterministically

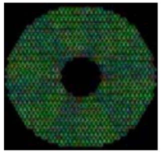
We applied this approach to matrix-vector multiplication and PDE solver problems that are at the heart of many control applications. PDEs are used to describe the system and matrix-vector multiplication is used to apply the model to sensor data in order to generate actuator data.

Matrix-Vector Multiplication in real-time



E-ELT Telescope

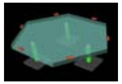
ESO (European Southern Observatory)
Five mirrors (M1 through M5)
National Instruments involvement:
- **Data Acquisition**
- **M1 – 42m primary segmented mirror**
- **M4 – adaptive mirror**



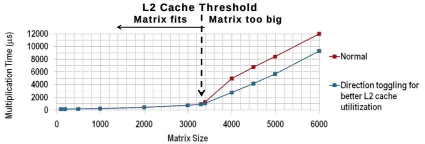
E-ELT Telescope: M1 Mirror

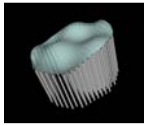
984 hexagonal segments
6 sensors/3 actuators per segment
1 ms control cycle

Math tricks to reduce problem to 3k x 3k symmetric matrix by 3k vector multiplication



New multiplication algorithm needed
750 µs (worst case)
Dell T7400 (2x2.6GHz 12MB L2 Quad Core Xeon)
Matrix restructured off-line in L1/L2 cache optimized sub-problems and loaded into the CPU caches
GPU (dual Tesla): 850 µs



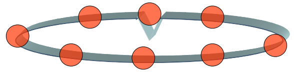


E-ELT Telescope: M4 Mirror

6000-9000 actuators
1ms control cycle
Math: 9k x 15k matrix by 15k vector multiplication

Both CPU bandwidth and cache size limited
➔distributed computation required
Custom FPGA-based communication protocol:
1.Deterministic communication
2.On-the-fly data recombination

Custom FPGA-based Communication Protocol

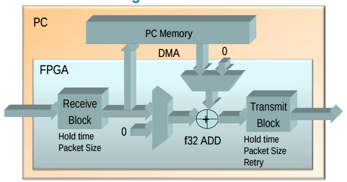


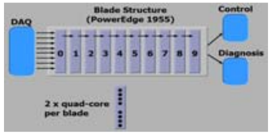
Network topology: ring
Network speed: up to 110 MB/s (cable dependent)
Configurable: RX and TX parameters, data sink, data source, data recombine
Development time: 2 weeks

Throughput/Latency for 80MB/s network setup

Packet length (f32s)	Throughput (MB/s)	First byte latency per node (µs)
8	46	1
16	57	1.5
32	65	2.5
64	70	4.5

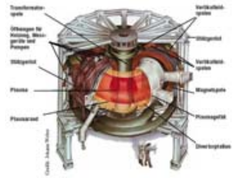
FPGA Block Diagram





Multi-blade setup for distributed calculation

(Non)Linear Elliptic PDE Solver



Tokamak Fusion Reactor

$$R\frac{\partial}{\partial R}\left(\frac{1}{R}\frac{\partial \psi}{\partial R}\right)+\left(\frac{\partial^2 \psi}{\partial Z^2}\right)=-\mu_o RJ(\psi)$$

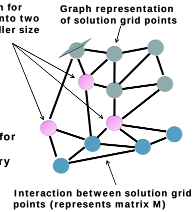
Grad-Shafranov PDE
Magnetohydrodynamics Shape Control

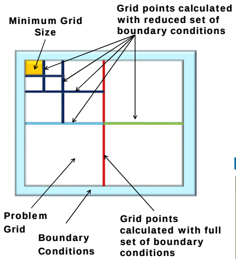
Currently: 3-12 ms, 39x63 grid
Goal: 1ms

Geometry-aware algorithm:
1. **Cast PDE into a set of equations:** $Mx - b = f(x)$
2. **Use iterative algorithm:** $x^{(n+1)} = M^{-1}\{f[x^{(n)}] + b\}$
3. **Reduce problem for each iteration by finding a set of points (pink) that, when calculated, exactly split the problem into two or more smaller independent sub-problems (blue and green)**
4. **Calculate the exact solution for the points using corresponding rows of M^{-1}**
5. **Repeat for each sub-problem**

Problem:
For a 1000x1000 grid, M dimensions are M(1,000,000 x 1,000,000)
Calculating M^{-1} directly may be impossible

Solution:
Solve the problem for the unit boundary vector using any method (for example, iteration over smaller subsets of the problem for which M-1 can be calculated directly). Values for the solution corresponding to the pink points represent elements in the M^{-1} rows corresponding to the unit boundary vector. Repeat the process for all unit vectors.





Geometry-aware algorithm applied to a rectangular PDE grid

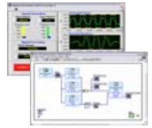
111x63 grid (6993 non-linear equations with 6993 unknowns)
7^{th} order RHS polynomial
One iteration: < 250 µs
10^{-5} error: 4 iterations starting from 0s
Total time for solution: <1 ms
Dell 7400T (2x2.6GHz Quad Core Xeons)

Linear PDE Benchmarks (µs)

Grid Size	Laplace			General Elliptic Liner PDE			
	Spectral	New Method (CPUs)		CPUs			
		1	8	1	2	4	8
31x31	115	39	73	40	32	34	72
63x63	495	96	84	100	66	61	85
127x127	2130	363	128	381	211	157	129
255x255	10321	1498	296	6073	2866	2408	308
319x319	16594	2589	459				1416
511x511	45748	10883	965	29195	15171	13873	11814

Tool

LabVIEW Graphical Development Environment



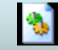
- Structured data flow programming
- Compiled graphical development
- Target desktop, mobile, industrial, and embedded
- Thousands of out-of-the box mathematics and signal processing routines
- Seamless connectivity to millions of I/O devices

Application Layers

Layer 1

Off-line calculations
Data distribution
CPU affinity feature for cache control (keep fixed data in cache)
Read sensor data, distribute across CPUs (machines). Collate and update outputs
Program FPGA
Call dll

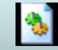
Layer 2



Math.dll

C code
Memory management
data alignment
Iterate over sub-problems

Layer 3



Math.dll

Assembly code
Hand coded SSE2 assembly for solving sub-problems
L2 & L1 cache optimized

Layer 4

Custom communication protocol
CPU of-load for data recombination
Fully hand-shaked, with retries
Slower nodes will gate data delivery
DMA for data access directly from memory

LabVIEW Targets