Reconfigurable Supercomputing with Scalable Systolic Arrays and In-Stream Control for Wavefront Genomics Processing

C. Pascoe (speaker), A. Lawande, H. Lam, A. George
NSF Center for High-Performance Reconfigurable Computing (CHREC), University of Florida

M. Herbordt
Department of Electrical and Computer Engineering, Boston University

Y. Sun, W. Farmerie
Interdisciplinary Center for Biotechnology Research (ICBR), University of Florida
Motivation

- Impending roadblock in health-sciences research
  - Current-generation sequencing technology capable of producing millions of sequences per day
  - Next-generation sequencing instruments poised to increase production more than 10-fold

- Conventional sequence-analysis processing has not kept pace with raw sequence production

**Objective:** Develop sustainable solution for increasing performance of sequence-analysis applications

**Approach:**
- Develop in-stream control approach for systolic array to maximize hardware acceleration
- Capitalize on reconfigurable computing at scale to achieve sustainable supercomputing performance
Presentation Outline

- Background
  - Sequential Score Calculation
  - Backtrack to Produce String Representation
  - Sequence-Alignment Acceleration
  - Common Hardware Acceleration Techniques
- Sequence-Alignment Acceleration Issues
- Scalable Systolic Arrays with In-Stream Control
- Demonstration of In-Stream Control: Case Studies
  - Needleman-Wunsch (NW)
  - Smith-Waterman (SW)
  - Needle-Distance (ND)
- Novo-G System Architecture
  - GiDEL PROCStar-III Boards
- NW/SW/ND Performance Evaluation on Novo-G
- Conclusions
- Q&A
Background: Sequence-Alignment Example

Sequence $S_1$: ACGGCTAACTG

Sequence $S_2$: GGCCCAACTGCGGGT

- Want to quantify similarity between $S_1$ and $S_2$
- Use NW or SW or other algorithms to align sequences

$$S(i, j) = \max \begin{cases} 
S(i - 1, j - 1) + Sub(i, j) \\
S(i - 1, j) + eog \\
H(i - 1, j) + e \\
S(i, j - 1) + eog \\
V(i, j - 1) + e 
\end{cases}$$

for $1 \leq i \leq i_{\text{max}}$, $1 \leq j \leq j_{\text{max}}$

$$H(i, j) = \max \begin{cases} 
S(i - 1, j) + og \\
H(i - 1, j) + eg 
\end{cases}$$

$$V(i, j) = \max \begin{cases} 
S(i, j - 1) + og \\
V(i, j - 1) + eg 
\end{cases}$$

$$S(0, i) = S(i, 0) = og + i \cdot eg$$

$$H(0, i) = V(i, 0) = og$$

$$Sub(i, j) = \text{penalty matrix}$$

Needleman-Wunsch Equations
## Sequential Score Calculation

Score Calculation --- match = 5, mismatch = -4, ogap = -10, egap = -0.5

<table>
<thead>
<tr>
<th>S1</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>A</th>
<th>C</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-10</td>
<td>-4</td>
<td>-14</td>
<td>-5.5</td>
<td>-6</td>
<td>-6.5</td>
<td>-7</td>
<td>-7.5</td>
<td>-8</td>
<td>-8.5</td>
<td>-9</td>
</tr>
<tr>
<td>G</td>
<td>-10.5</td>
<td>-14</td>
<td>-8</td>
<td>-9</td>
<td>-0.5</td>
<td>-10</td>
<td>-10.5</td>
<td>-11</td>
<td>-11.5</td>
<td>-12</td>
<td>-12.5</td>
</tr>
<tr>
<td>C</td>
<td>-11</td>
<td>-14.5</td>
<td>-9</td>
<td>-12</td>
<td>-10.5</td>
<td>4.5</td>
<td>-5.5</td>
<td>-6</td>
<td>-6.5</td>
<td>-6.5</td>
<td>-16</td>
</tr>
<tr>
<td>C</td>
<td>-11.5</td>
<td>-15</td>
<td>-9.5</td>
<td>-13</td>
<td>-11</td>
<td>-5.5</td>
<td>0.5</td>
<td>-9.5</td>
<td>-10</td>
<td>-1.5</td>
<td>-10.5</td>
</tr>
<tr>
<td>C</td>
<td>-12</td>
<td>-15.5</td>
<td>-10</td>
<td>-13.5</td>
<td>-11.5</td>
<td>-6</td>
<td>-9.5</td>
<td>-3.5</td>
<td>-13.5</td>
<td>-6</td>
<td>-5.5</td>
</tr>
<tr>
<td>A</td>
<td>-12.5</td>
<td>-17</td>
<td>-14</td>
<td>-12</td>
<td>-6.5</td>
<td>-10</td>
<td>-5.5</td>
<td>1.5</td>
<td>-8.5</td>
<td>-9</td>
<td>-9.5</td>
</tr>
<tr>
<td>A</td>
<td>-13</td>
<td>-17</td>
<td>-11</td>
<td>-21</td>
<td>-12.5</td>
<td>-7</td>
<td>-10.5</td>
<td>-5</td>
<td>-0.5</td>
<td>-3.5</td>
<td>-12.5</td>
</tr>
<tr>
<td>C</td>
<td>-13.5</td>
<td>-17</td>
<td>-2.5</td>
<td>-12.5</td>
<td>-13</td>
<td>-7.5</td>
<td>-11</td>
<td>-14.5</td>
<td>-9</td>
<td>4.5</td>
<td>-6.5</td>
</tr>
<tr>
<td>T</td>
<td>-14</td>
<td>-17.5</td>
<td>-12.5</td>
<td>-6.5</td>
<td>-13.5</td>
<td>-8</td>
<td>-2.5</td>
<td>-12.5</td>
<td>-13</td>
<td>-5.5</td>
<td>9.5</td>
</tr>
<tr>
<td>G</td>
<td>-14.5</td>
<td>-18</td>
<td>-13</td>
<td>-7.5</td>
<td>-1.5</td>
<td>-8.5</td>
<td>-12</td>
<td>-6.5</td>
<td>-16.5</td>
<td>-6</td>
<td>-0.5</td>
</tr>
<tr>
<td>C</td>
<td>-15</td>
<td>-18.5</td>
<td>-13</td>
<td>-17</td>
<td>-11.5</td>
<td>3.5</td>
<td>-6.5</td>
<td>-16</td>
<td>-10.5</td>
<td>-6.5</td>
<td>-1</td>
</tr>
<tr>
<td>G</td>
<td>-15.5</td>
<td>-19</td>
<td>-22.5</td>
<td>-8</td>
<td>-12</td>
<td>-6.5</td>
<td>-0.5</td>
<td>-10.5</td>
<td>-20</td>
<td>-7</td>
<td>-1.5</td>
</tr>
<tr>
<td>G</td>
<td>-16</td>
<td>-19.5</td>
<td>-23</td>
<td>-17.5</td>
<td>-3</td>
<td>-7</td>
<td>-10.5</td>
<td>-4.5</td>
<td>-14.5</td>
<td>-7.5</td>
<td>-2</td>
</tr>
<tr>
<td>G</td>
<td>-16.5</td>
<td>-20</td>
<td>-23.5</td>
<td>-18</td>
<td>-12.5</td>
<td>-7</td>
<td>-11</td>
<td>-14.5</td>
<td>-8.5</td>
<td>-8</td>
<td>-2.5</td>
</tr>
<tr>
<td>T</td>
<td>-17</td>
<td>-20.5</td>
<td>-24</td>
<td>-27.5</td>
<td>-22</td>
<td>-16.5</td>
<td>-2</td>
<td>-12</td>
<td>-18.5</td>
<td>-8.5</td>
<td>-3</td>
</tr>
</tbody>
</table>
### Backtrack to Produce String Representation

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>A</th>
<th>C</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>G</td>
<td>V</td>
<td>D</td>
<td>H</td>
<td>D</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>G</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>V</td>
<td>D</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>H</td>
<td>H</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>H</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>A</td>
<td>V</td>
<td>D</td>
<td>H</td>
<td>D</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>A</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>H</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>H</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>H</td>
</tr>
<tr>
<td>T</td>
<td>V</td>
<td>D</td>
<td>V</td>
<td>D</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>H</td>
<td>H</td>
<td>V</td>
<td>D</td>
</tr>
<tr>
<td>G</td>
<td>V</td>
<td>D</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>H</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>C</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>V</td>
<td>D</td>
<td>H</td>
<td>D</td>
<td>D</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>G</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>H</td>
<td>D</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>G</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>H</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>G</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>V</td>
<td>V</td>
<td>D</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>V</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>H</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
</tbody>
</table>
**Sequence-Alignment Acceleration**

Score Calculation --- match = 5, mismatch = -4, ogap = -10, egap = -0.5

<table>
<thead>
<tr>
<th>S2</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>A</th>
<th>C</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>-10</td>
<td>-10.5</td>
<td>-11</td>
<td>-11.5</td>
<td>-12</td>
<td>-12.5</td>
<td>-13</td>
<td>-13.5</td>
<td>-14</td>
<td>-14.5</td>
<td>-15</td>
</tr>
<tr>
<td>G</td>
<td>-10.5</td>
<td>-14</td>
<td>-8</td>
<td>-9</td>
<td>-0.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-11</td>
<td>-14.5</td>
<td>-9</td>
<td>-12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-11.5</td>
<td>-15</td>
<td>-9.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-12</td>
<td>-15.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>-12.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>-13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-13.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-14.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-15.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-16.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Common Hardware Acceleration Techniques

- Construct **systolic array** of processing elements (PEs) to exploit parallelism
- Each **PE** responsible for calculating single column of scoring matrix
- $O(X \cdot Y)$ time complexity in software reduced to $O(X+Y)$ in hardware

![Diagram of systolic array](image)
**Sequence-Alignment Acceleration Issues**

- **Maximum possible** performance directly proportional to:
  - Number of PEs
  - Operating clock frequency
  (e.g. 100 PEs x 100MHz = 10 GCUPS)

- **Achieved** performance can be much less due to overhead:
  - Input data manipulation
  - Input data transfer to FPGA
  - FPGA initialization and setup
  - Datapath initialization
  - PE configuration
  - Pipeline latency
  - Handshaking control between FPGA/CPU
  - Output data transfer to CPU
  - Output data manipulation
  - Others

- **Design goals:**
  - Eliminate or overlap overhead with useful work
  - Minimizing PE area to maximize the number of PEs per FPGA
Scalable Systolic Arrays with In-Stream Control

**Worst-case Performance → Low-area Overhead**

\[ N \text{ (setup time + pipeline latency + CPU/FPGA handshake + PE configuration time + time to record results + time to process streamed sequence)} \]

**Best-case Performance → High-area Overhead**

pipeline latency + \( N \) (time to process streamed sequence)

**With In-Stream Control**

**Near-best-case Performance → Low-area Overhead**

pipeline latency + \( M \) (PE configuration + 2 clock cycles) + \( N \) (time to process streamed sequence + 2 clock cycles)

- Replace complex state machines and other supporting logic with control data inserted directly into data stream
  - Signal state transitions
  - Trigger complex actions on application data
  - Limit wasted cycles
  - Reduce PE area
  - Increase hardware utilization

*Complex-controller performance with simple-controller overhead*
In-Stream Control: Needleman-Wunsch

- **Needleman-Wunsch (NW)** used for global sequence alignment
- Conventional systolic-array architecture
  - With simple PEs and simple control
- Input sequences consist of standard nucleotide alphabet
  - (A, C, G, T, N)
- 3-bit encoding:
  - Add 3 additional control characters (total of 8 characters):
    - (L) load query, (R) reset to initial conditions, (P) push results
  - Special 2-character sequence “PN” used to signal end of stream
Example set of short sequences:  
{ACGT, TTG, ACNG, CNTG}

Assume it is desired to perform NW between each unique sequence pair in set:  
(i.e. 1-2, 1-3, 1-4, 2-3, 2-4, and 3-4)

Stream encoding

“LACGTRTTGPRACNGPRCNTGPLTTGRACNGPRCNTGPLACNGRCNTGPN”

After (50 + pipeline length) FPGA clock cycles, the six results are ready for retrieval when CPU has free cycles
In-Stream Control for NW: Summary

- Introduces notion of local control
  - Control characters used as framing characters for each loaded or streamed sequence
  - Simultaneously:
    - PEs in pipeline between an ‘L’ and ‘R’ can be reconfiguring
    - Earlier group of PEs between an ‘R’ and ‘P’ can be processing a query
    - Even earlier group is doing something else

- Simplifies control:
  - Meaningful outputs result from correct sequence of characters passing through systolic array
  - No need for control signals from controller and CPU

- Benefit:
  - Limits PE idle periods
  - Reduces configuration overhead between runs
  - Reduces PE area

- Cost?
  - Only two additional clock cycles of overhead per sequence to process framing characters
In-Stream Control: Smith-Waterman

- **Smith-Waterman (SW)** used for local sequence alignment
- Use NW control structure as basis then modify to add functionalities
  - Switching between query and database stream
    - Requires overloaded ‘L’ character to toggle between streams
  - Loading multiple queries against a single streamed database
    - Introduce concept of “push window” and add extra bit to CPU stream encoding for controlling query start and query extend modes
  - Providing traceback (string representation of alignment)
    - No changes to encoding scheme, modification to supporting software control
  - Extending queries across multiple FPGAs
    - No changes to encoding scheme, modification to supporting software control

*Optimized for Short Sequences vs. Large Database*
**In-Stream Control for SW: Simple Example**

- Example set of short sequences: \{ACGT, TTG, ACNG, CNTG\}
- Assume it is desired to perform SW between each sequence and a large database
- Database Stream (3bit) Encoding:
  "RAC…GTPPPPL"
- Query Stream (4bit) Encoding:
  "(0L)(1A)(0C)(0G)(0T)(1T)(0T)(0G)(1A)(0C)(0N)(0G)(1C)(0N)(0T)(0G)(0R)(0L)(0P)(0N)"

After (20 + database length + pipeline length) FPGA clock cycles, the 4 results are ready for retrieval when CPU has free cycles.
In-Stream Control: Needle-Distance

- Needle-Distance (ND) used for calculating optimal pairwise distance

Original Software:
1. Calculate NW score
2. Generate string representation
3. Calculate custom normalized edit distance

Hardware:
- Score and distance calculations performed in parallel
- Eliminate computation of string representation

ND PEs implemented by adding distance calculation modules to PEs of NW

```
A C G G C T _ A A C T G _ _ _ _ _ _
_ _ G G C C C A A C T G C G G G T
```

Edit distance:

\[ 0 + 0 + 0 + 0 + 1 + 1 + 0 + 0 + 0 + 0 + 0 = 2 \]

Total distance:

\[ 0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 0 = 10 \]

Normalized edit distance:

\[ \frac{2}{10} = 0.200000 \]
Novo-G System Architecture

- **24+1 Linux servers in cluster**
  - 24 compute servers (2 boards/server)
  - 1 head-node server for management
  - 20 Gb/s non-blocking InfiniBand
  - 1 Gb/s Ethernet
  - 26 (24+2) quad-core Xeons
  - Max. system power of ~8 KW

- **48 quad-FPGA boards**
  - GiDEL PCIe x8 PROCStar-III
    - Embedded-style boards, for both HPEC- & HPC-oriented research
  - 4¼ GB DDR2 attached to each FPGA
    - ~ 1 TB total RAM in Novo-G

- **192 Stratix-III E260 FPGAs**
  - Each Altera FPGA with 768 18x18 multipliers, 254K logic elements, 204K registers, & max. power ~18W


“Novo” is Latin, "to make anew, refresh, revive, change, alter," essence of RC; “G” is for Genesis or Green. 😊
GiDEL PROCStar-III Board (one of 48)

**Altera Stratix-III E260 FPGA**
- 254,400 Logic Elements
- 768 multipliers (18x18)
- 14,688 Kbits of embedded memory
- 50% less power than Stratix-II
- 65nm technology

**GiDEL PROCStarIII Board**
- Typical frequencies: 100-325MHz
- DMA channels: 32
- DDR2 module slots: 8

2GB = 4GB DDR2 RAM per FPGA

**25.6 Gb/s inter-FPGA bandwidth**

**JTAG for SignalTap debug**

**NOTE: GiDEL developed multiboard interconnect; stream apps across multiple PS-III boards within servers**
Needleman-Wunsch Performance Evaluation on Novo-G

Baseline: 192 \cdot 2^{25}, length 850 Sequence Comparisons
Software Runtime: 11,026 CPU hours on 2.4GHz Opteron

<table>
<thead>
<tr>
<th># FPGAs</th>
<th>Runtime (sec)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47,616</td>
<td>833</td>
</tr>
<tr>
<td>4</td>
<td>12,014</td>
<td>3,304</td>
</tr>
<tr>
<td>96</td>
<td>503</td>
<td>78,914</td>
</tr>
<tr>
<td>128</td>
<td>391</td>
<td>101,518</td>
</tr>
<tr>
<td>192 (est.)</td>
<td>270</td>
<td>147,013</td>
</tr>
</tbody>
</table>

Scaling performance with varying number of FPGAs under optimal input conditions

Number of Sequence Comparisons
Performance of single FPGA under varying input conditions

NW GCUPS Achieved/Max
17,240/20,400 = 85%
SW/ND Performance Evaluation on Novo-G

Baseline: Human X Chromosome v 19200, length 650 Seqs
Software Runtime: 5,481 CPU hours on 2.4GHz Opteron

<table>
<thead>
<tr>
<th># FPGAs</th>
<th>Runtime (sec)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23,846</td>
<td>827</td>
</tr>
<tr>
<td>4</td>
<td>5,966</td>
<td>3,307</td>
</tr>
<tr>
<td>96</td>
<td>250</td>
<td>78,926</td>
</tr>
<tr>
<td>128</td>
<td>188</td>
<td>104,955</td>
</tr>
<tr>
<td>192 (est.)</td>
<td>127</td>
<td>155,366</td>
</tr>
</tbody>
</table>

SW GCUPS Achieved/Max 15,158/15,600 = 97%

Baseline: 192·2^{24}, length 450 Distance Calculations
Software Runtime: 11,673 CPU hours on 2.4GHz Opteron

<table>
<thead>
<tr>
<th># FPGAs</th>
<th>Runtime (sec)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13,522</td>
<td>3,108</td>
</tr>
<tr>
<td>4</td>
<td>3,429</td>
<td>12,255</td>
</tr>
<tr>
<td>96</td>
<td>144</td>
<td>291,825</td>
</tr>
<tr>
<td>128</td>
<td>118</td>
<td>356,125</td>
</tr>
<tr>
<td>192 (est.)</td>
<td>77</td>
<td>545,751</td>
</tr>
</tbody>
</table>

ND GCUPS Achieved/Max 8,471/10,800 = 78%
Conclusions

- Developed **in-stream control** approach for systolic array to maximize hardware acceleration
- Capitalized on **reconfigurable computing at scale** to achieve **sustainable supercomputing performance**
- Demonstrated **performance and sustainability** results on Novo-G:

<table>
<thead>
<tr>
<th># FPGAs</th>
<th>Runtime (sec)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47,616</td>
<td>833</td>
</tr>
<tr>
<td>4</td>
<td>12,014</td>
<td>3,034</td>
</tr>
<tr>
<td>96</td>
<td>503</td>
<td>78,914</td>
</tr>
<tr>
<td>128</td>
<td>391</td>
<td>101,518</td>
</tr>
<tr>
<td>192 (est.)</td>
<td>270</td>
<td>147,013</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># FPGAs</th>
<th>Runtime (sec)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23,846</td>
<td>827</td>
</tr>
<tr>
<td>4</td>
<td>5,966</td>
<td>3,307</td>
</tr>
<tr>
<td>96</td>
<td>250</td>
<td>78,926</td>
</tr>
<tr>
<td>128</td>
<td>188</td>
<td>104,955</td>
</tr>
<tr>
<td>192 (est.)</td>
<td>127</td>
<td>155,366</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># FPGAs</th>
<th>Runtime (sec)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13,522</td>
<td>3,108</td>
</tr>
<tr>
<td>4</td>
<td>3,429</td>
<td>12,255</td>
</tr>
<tr>
<td>96</td>
<td>144</td>
<td>291,825</td>
</tr>
<tr>
<td>128</td>
<td>118</td>
<td>356,125</td>
</tr>
<tr>
<td>192 (est.)</td>
<td>77</td>
<td>545,751</td>
</tr>
</tbody>
</table>

Results on Novo-G for NW (left), SW (Center), and ND (Right). Each table shows scaling performance with varying number of FPGAs under optimal input conditions.

Estimated performance on Novo-G comparable or better than biggest supercomputers on www.Top500.org

Jaguar @ ORNL: **224,162 cores** – 2.4 GHz Hexacore Opterons; **6.95 MW**

Roadrunner @ LANL: **122,400 cores** – 3.2 GHz Cells + 1.8GHz Opterons; **2.35 MW**

Novo-G Power 8 KW Max.
Thanks for listening!

Questions
References


[7] ESPRIT download, plaza.ufl.edu/sunyijun/ESPRIT.htm