



# **HPC-BLAST**

## **Scalable Sequence Analysis for the Intel® Many Integrated Core Future**

**Dr. R. Glenn Brook & Shane Sawyer**  
**Joint Institute For Computational Sciences**  
**University of Tennessee, Knoxville**

**Dr. Bhanu Rekepalli**  
**BioTeam, Inc.**

# HPC-BLAST Development

- **HPC-BLAST** is a high-throughput, highly parallel version of the **NCBI Basic Local Alignment Search Tool**.
  - Designed for use with the Intel® Many Integrated Core architecture
- The **NCBI BLAST** algorithm is used extensively in bioinformatics for comparative sequence analysis.
- **HPC-BLAST** is scalable, employing **MPI** and **OpenMP**.
  - Executes **natively** on Xeon processors or Xeon Phi coprocessors
  - Executes **symmetrically** across Xeon processors and Xeon Phi coprocessors
- The **development** of HPC-BLAST extensively used **tools**.
  - Rogue Wave TotalView
  - Allinea DDT
  - Intel VTune and Inspector
  - 2013 and 2015 Intel C++ compilers

# The NCBI-BLASTp algorithm: Overview

[1] Concatenate queries.

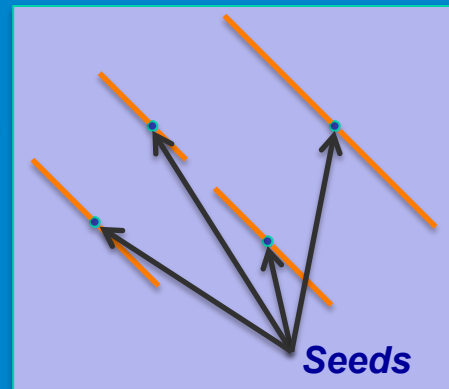


Word	Offset
YVE	13
VED	14
EDI	15
DIY	16
IYR	17

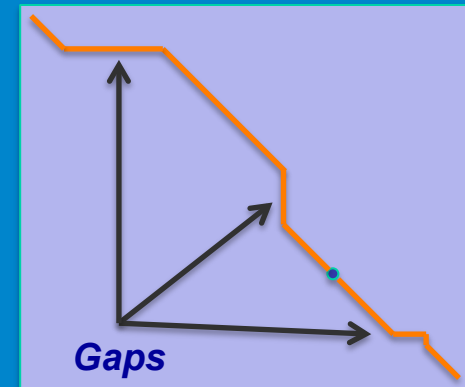
Word	Offset
YVE	13
VED	14
EDI	15
DIY	16
IYR	17

[3] Mark matches (seeds) in the lookup table.

[4] Perform ungapped alignment.



[5] Perform gapped alignment allowing for blanks to be inserted.



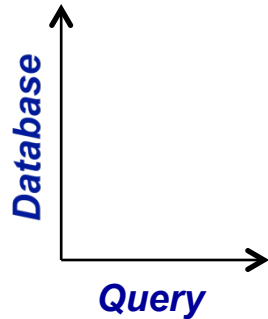
[2] Build a lookup table.

Query: YVEDIYRDH GALPPMS  
 VED+ R+HG ++ +  
 Subject: RVEDDCRCHGV TMMRR

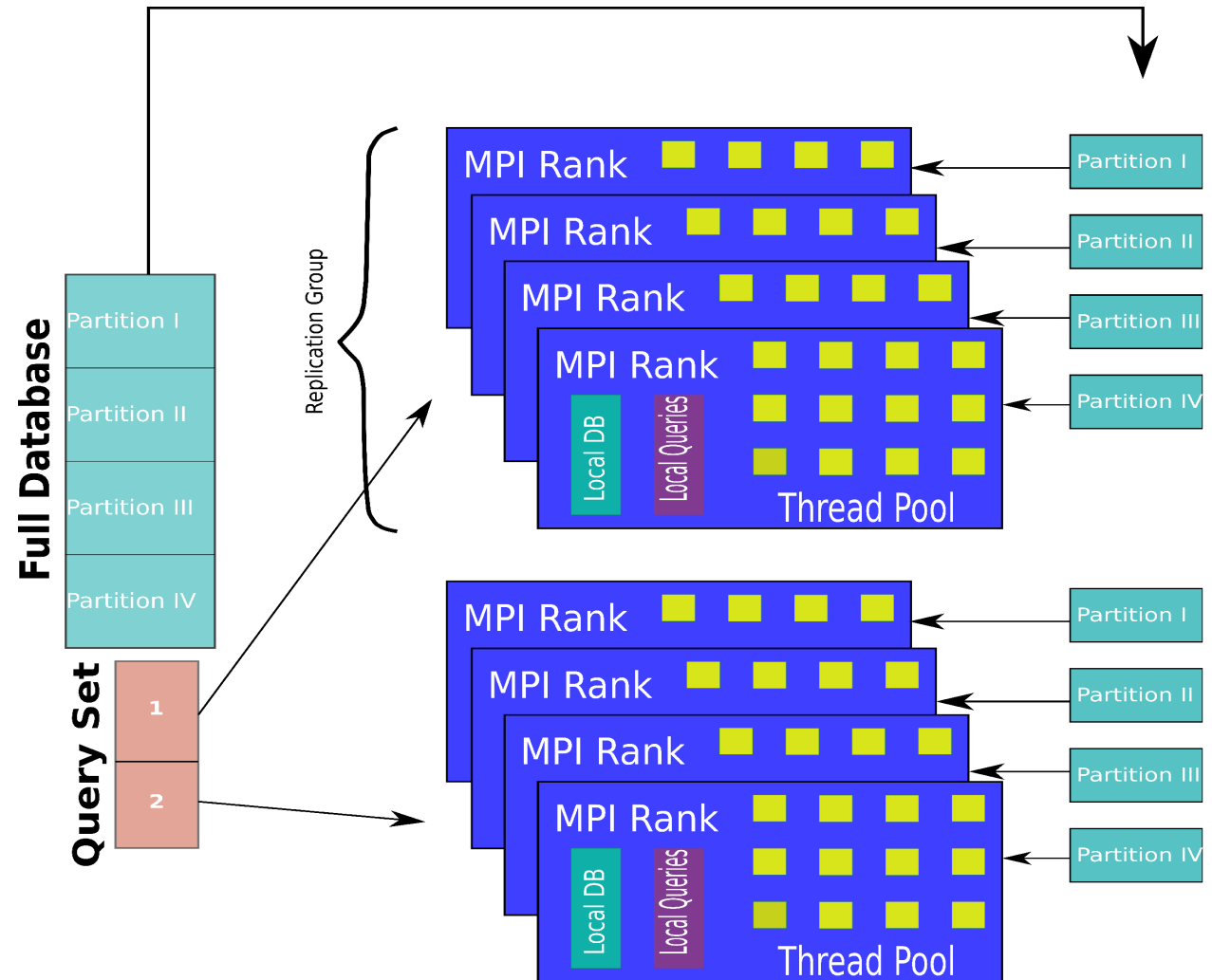
[6] Report high-scoring matches.

# HPC-BLAST Approach – MPI Level

Axes of Decomposition

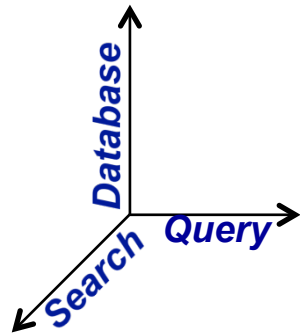


- A **replication group** consists of one or more MPI ranks that collectively contain a **complete copy** of the subject database.
- **Multiple replication groups** allow for distribution of the **query database**.
- **Multiple MPI ranks** within a replication group allow for distribution of the **search database**.

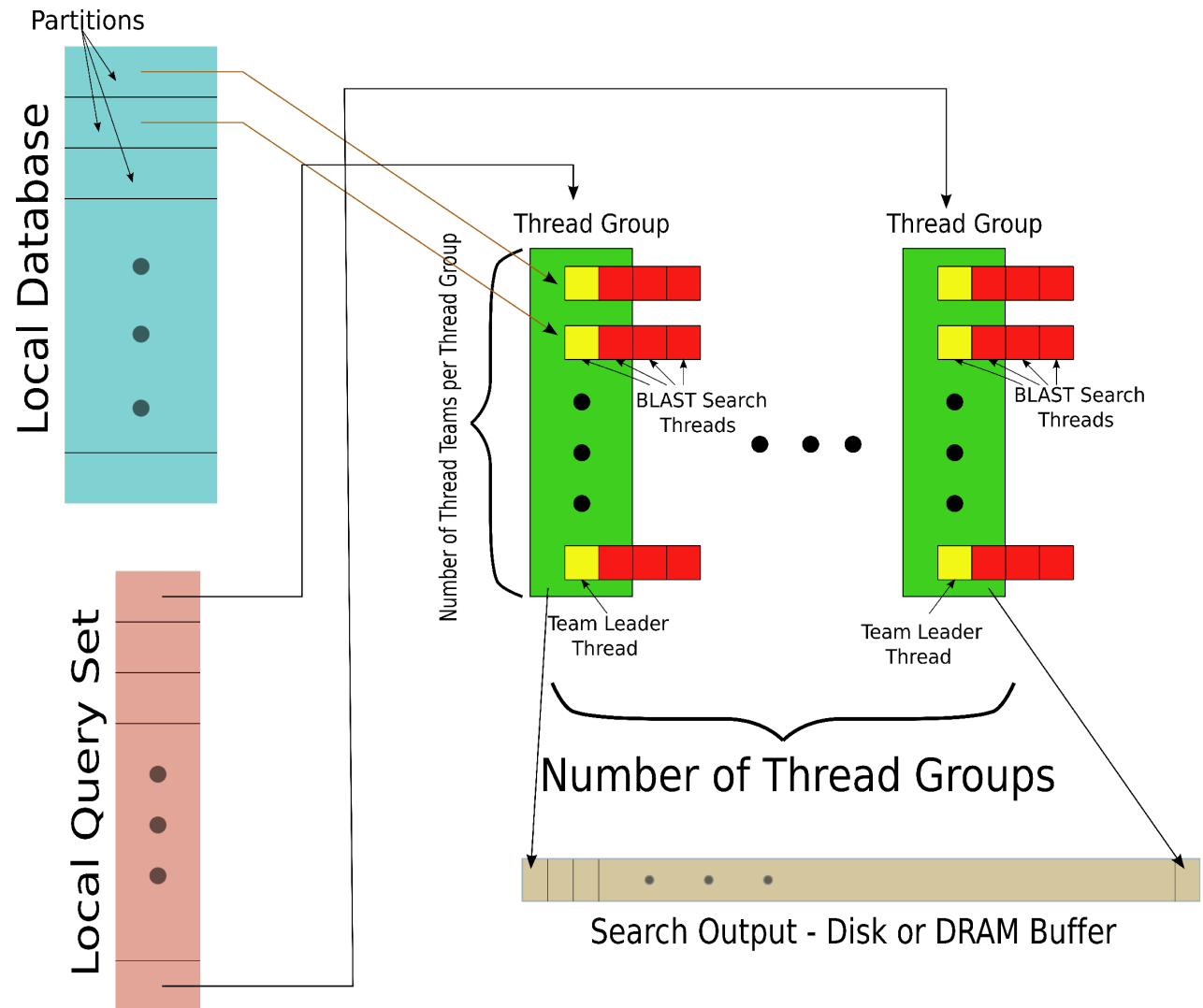


# HPC-BLAST Approach – Thread Level

## Axes of Decomposition



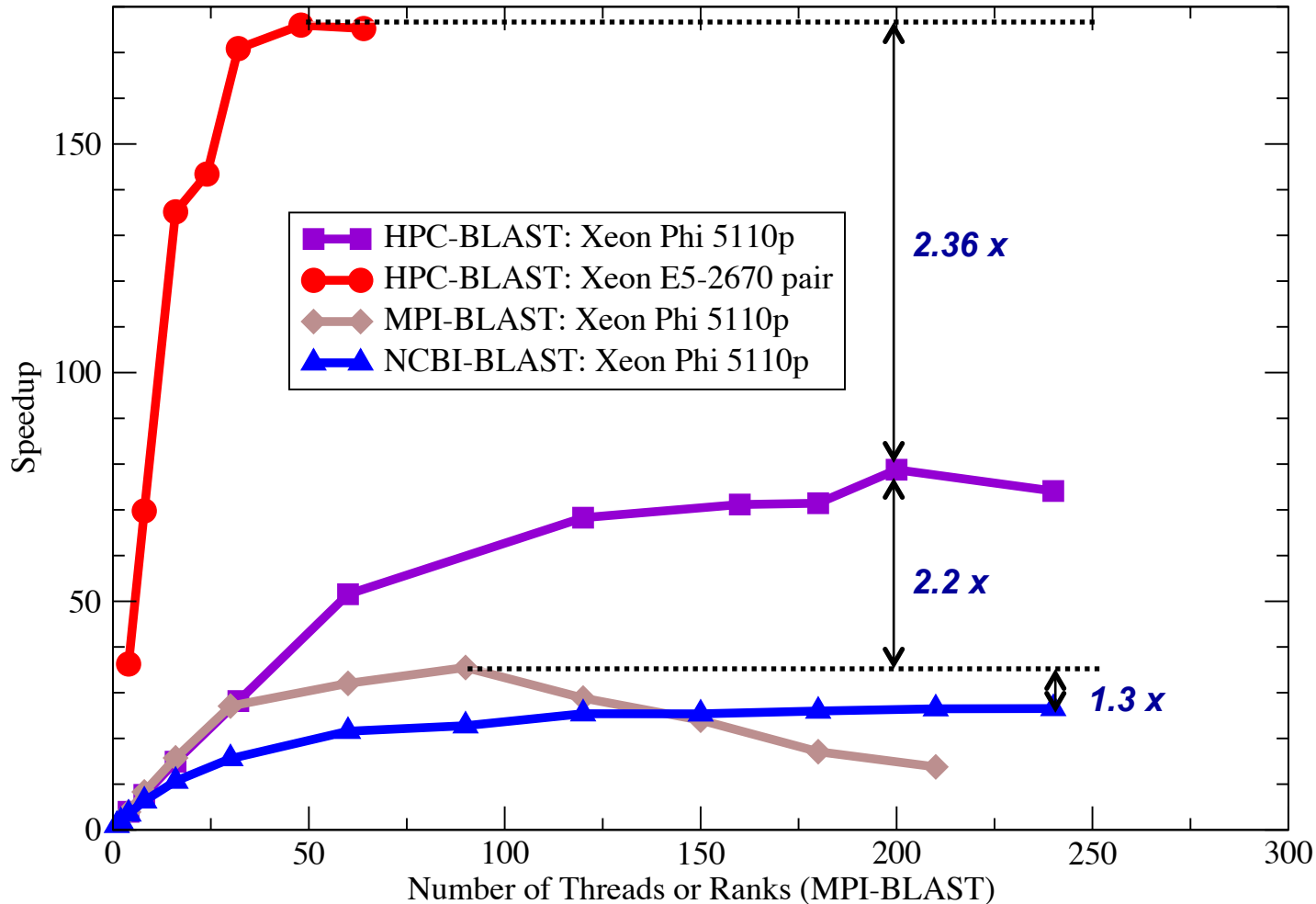
- A **thread group** consists of one or more **leader threads** that collectively contain a **complete copy** of the subject database.
- **Multiple thread groups** allow for distribution of the **query database**.
- **Multiple leader threads** within a thread group allow for distribution of the **search database**.
- **Multiple search threads** provide threads to the **core BLAST** search algorithm.



# Performance: Strong Scaling

## Relative Performance: Comparison of BLAST implementations

Xeon Phi 5110p v.s. Xeon E5-2670 pair



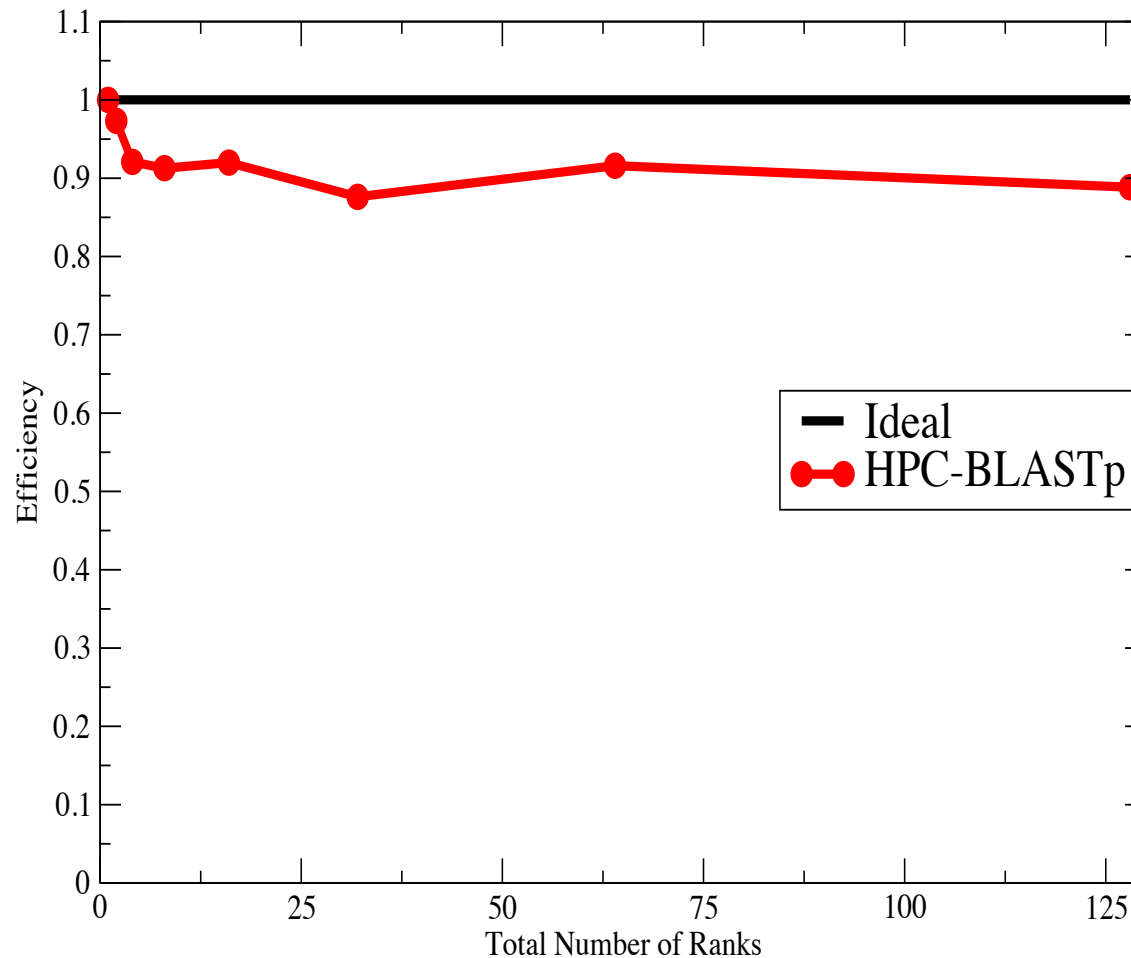
- Single Xeon Phi performance compared to pair of Xeon E5-2670
- Speedup relative to single NCBI thread on Phi
- 4 HPC-BLASTp ranks on Phi
- 1 HPC-BLASTp rank on Xeons

• Subject database: nr.01 partition; Query database: 1065 random sequences from nr.01  
• BLAST+ 2.2.28; mpiBLAST 1.6.0

# Performance: Weak Scaling

## HPC-BLASTp Weak Scaling

Xeon Phi 5110p, 240 Threads per Phi

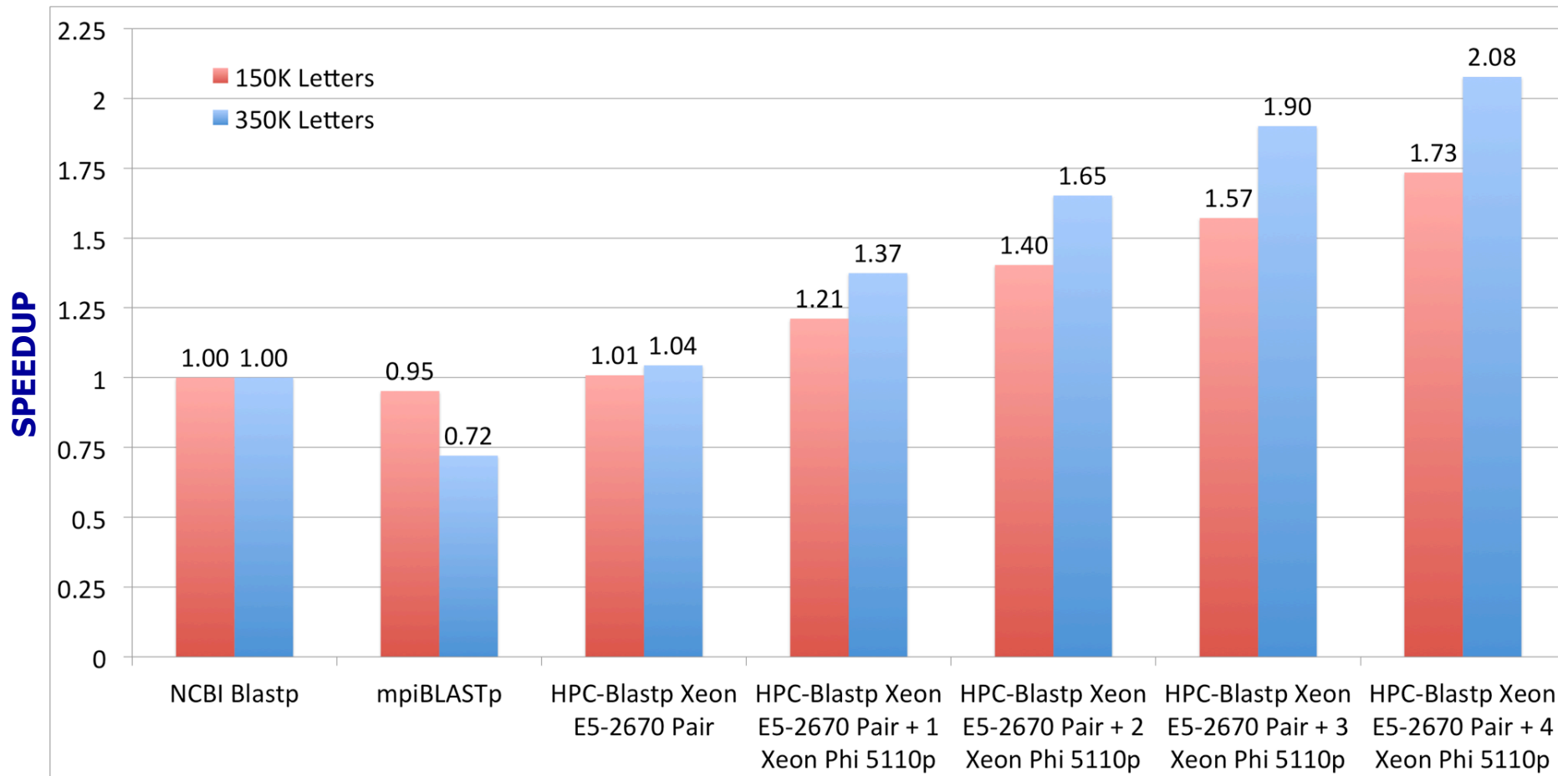


- 32 nodes
- 4 coprocessors per node
- 1 MPI rank per coprocessor
- 4 threads per coprocessor core
- Identical workload on each rank

- Subject database: nr.01 partition; Query database: 500 sequences of length 350 amino acids
- BLAST+ 2.2.28

# Performance: Throughput

The HPC-BLAST approach enables **scientific impact** with impressive throughput.



• ***Xeon + Xeon Phi provides improved performance per dollar for sufficiently large searches of the entire nr database from NCBI***

- Subject database: nr (entire database)
- Query database sampled from subjects: 444 (red) and 1065 (blue)
- BLAST+ 2.2.28; mpiBLAST 1.6.0



# Observations, Insights, & Future Work

- **Different sequence analysis problems exhibit different performance characteristics.**
- **Problem decomposition impacts performance pattern.**
- **Standard BLAST algorithm does not vectorize well.**
- **Flexibility in expressing parallelism at all levels is crucial for tuning performance.**
- **VTune is very helpful for performance analysis; Inspector is useful in finding race conditions.**
- **Future work:**
  - Implement scalable, parallel I/O
  - Improve load balancing – currently 30% - 40% imbalanced
  - Refactor to utilize SIMD operations while matching canonical output – challenging to maintain compatibility with heuristics

# THANK YOU

**For further information, please contact:**

**R. Glenn Brook, Ph.D.**

Chief Technology Officer, Joint Institute for Computational Sciences  
Director, Application Acceleration Center of Excellence  
Co-Director, UT Intel Parallel Computing Center @ JICS  
The University of Tennessee, Knoxville

**[Glenn-Brook@Tennessee.edu](mailto:Glenn-Brook@Tennessee.edu)**