Parallel k-means Clustering of Geospatiotemporal Data Sets Using Manycore CPU Architectures

Richard Tran Mills[α], Vamsi Sripathi[γ], Jitendra Kumar[β], Sarat Sreepathi[β], Forrest M. Hoffman[β], William W. Hargrove[δ]

[α]Argonne National Laboratory, [β]Oak Ridge National Laboratory, [γ]Intel Corporation, [δ]USDA Forest Service Southern Research Station

1. Introduction

- The increasing availability of high-resolution geospatiotemporal data sets from sources such as observatory networks, remote sensing platforms, and computational Earth system models has opened new possibilities for knowledge discovery and mining of ecological data sets fused from disparate sources.
- Traditional algorithms and computing platforms are impractical for the analysis and synthesis of data sets of this size; however, new algorithmic approaches that can effectively utilize the complex memory hierarchies and the extremely high levels of available parallelism in state-of-the-art high-performance computing platforms can enable such analysis.
- We describe pKluster, an open-source tool we have developed for accelerated k-means clustering of geospatiotemporal data. pKluster supports distributed-memory parallelism and can effectively utilize state-of-the art multi- and manycore processors, such as the second-generation Intel Xeon Phi ("Knights Landing") processor, as well as GPGPUs.
- We examine some practical applications of pKluster to the climate, remotely-sensed vegetation phenology, and LiDAR data sets and speculate on some of the other applications that such scalable analysis methods may enable.

2. Scalable k-means Clustering with pKluster

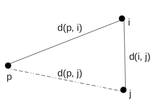
2.1 The pKluster distributed memory parallel k-means code

- Originally developed in 1996–1997 for use on the Stone Soupercomputer, a very early Beowulf-style cluster constructed entirely out of surplus parts (see "The Do-It-Yourself Supercomputer", *Scientific American*, 265 (2), pp. 72-79, 2001.)

- Because of extreme heterogeneity of the cluster, a master-slave parallel programming paradigm was used, as this provided excellent dynamic load-balancing.
- On modern, homogeneous machines, the master-slave paradigm may be less efficient than a fully-distributed, masterless approach.
  - We have explored the masterless approach in a prototype rewrite of the code.
  - We work with the master-slave version here, because some techniques described below introduce load imbalance even on homogeneous machines.
- When pKluster was initially written, on-node parallelism was virtually nonexistent on commodity PCs; the focus was purely on distributed-memory parallelism.
- Features:
  - Planned open-source release under the Apache License 2.0.
  - Runs on any machine (or cluster) with C89 (or higher) C compiler and an MPI implementation.
  - Option to improve cluster quality by moving or "warping" clusters that become empty to locations in data space where points that are farthest from their current cluster centroids reside.
  - Implements "accelerated" k-means algorithm.
  - Optimizations for manycore CPU and GPGPU systems.
  - Coming soon: Support for clustering observation vectors with many zero entries (e.g., species occurrence data).

2.2 "Accelerated" k-means Algorithm

- For very large datasets and/or cases when the number of clusters k is large, straightforward implementation of k-means proves too expensive, even when using many compute nodes.
- We "accelerate" the k-means process using two techniques described by Phillips (doi:10.1109/IGARSS.2002.1026202):
  - Use the triangle inequality to eliminate unnecessary point-to-centroid distance computations based on the previous cluster assignments and the new inter-centroid distances.
  - Reduce evaluation overhead by sorting inter-centroid distances so that new candidate centroids c_j are evaluated in order of their distance from the former centroid c_i. Once the critical distance $2d(p, c_i)$ is surpassed, no additional evaluations are needed, as the nearest centroid is known from a previous evaluation.



$d(i,j) \le d(p,i) + d(p,j)$
$d(i,j) - d(p,i) \le d(p,j)$
if $d(i,j) \ge 2d(p,i)$:
$d(p,j) \ge d(p,i)$
without calculating the distance $d(p,j)$

Figure 2: *The triangle inequality is used to eliminate unnecessary distance calculations.*

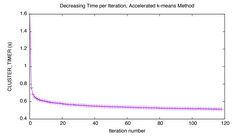


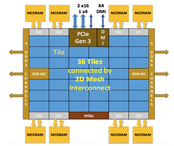
Figure 3: *Clustering the GSMNP LiDAR dataset from section 4 for $k = 2000$ with the accelerated k-means algorithm on the* BDW *system. Time for each iteration decreases as the accelerated algorithm is able to avoid many distance comparisons.*

- We also improve cluster quality by moving or "warping" clusters that become empty to locations in data space where points that are farthest from their current cluster centroids reside.

3. Manycore Computing Architectures

- In recent years, the number of CPU cores and hardware threads has been dramatically increasing.
- This is true on both standard server processors, as well as "manycore" processors such as the Intel Xeon Phi.
- There is also increasing reliance on data parallelism/fine-grained parallelism.
  - Second-generation Intel Xeon Phi processors have 512-bit vectors/AVX512 instructions; Skylake and Kaby-Lake Intel Xeon processors also support AVX512.
  - Slightly older Intel Xeon processors (Haswell and Broadwell generations) have 256-bit vector registers and support AVX2 instructions.
- Here, we adapt pKluster to Intel Xeon processors from recent generations and the Many Integrated Core (MIC) architecture of the second-generation "Knights Landing" (KNL) Intel Xeon Phi.

Second generation "Knights Landing" Intel Xeon Phi processors



- Available as standalone, self-boot CPU—no offload bottleneck; binary compatible with Intel Xeon Processors
- Up to 36 tiles interconnected via 2D mesh
- Tile: 2 cores + 2 VPU/core + 1 MB L2 cache
- Core: Silvermont-based, 4 threads per core, out-of-order execution
- Dual issue; can saturate both VPUs from a single thread
- 512 bit SIMD lanes, AVX512 vector instructions
- High bandwidth memory (MCDRAM) on package: 490+ GB/s bandwidth on STREAM triad

4. Benchmarking Setup and Baseline Performance

4.1 Benchmarking Platforms

- We use three hardware platforms: A 68-core KNL node, and dual-socket systems with "EP" versions of the current (Skylake, SKX) and previous (Broadwell, BDW) generation Intel Xeon server processors.
- All have similar power envelopes, so, from a power efficiency standpoint, a comparison of their performance is appropriate.
- SKX has some features similar to KNL that are new to the Xeon line: it has 512-bit vector registers and uses a variant of the AVX-512 instruction set, and it uses a mesh-on-die interconnect instead of the ring architecture used in previous Xeon generations.
- Both Xeon platforms deliver much higher per-thread performance than does KNL. Therefore, in general it is critically important for applications to possess sufficient parallel scalability to use most or all of the available cores or hardware threads in order to deliver competitive performance on KNL.

Table 1: *Characteristics of the computing platforms used for performance benchmarking in this study.*

	Intel Xeon E5-2697 v4	Intel Xeon Gold 6148	Intel Xeon Phi 7250
Code Name	Broadwell (BDW)	Skylake (SKX)	Knights Landing (KNL)
Sockets	2	2	1
Cores	36	40	68
Threads	72	80	272
CPU clock	2.3 GHz	2.4 GHz	1.4 GHz
High-bandwidth memory	-	-	16 GB
DRAM	128 GB @ 2400 MHz	192 GB @ 2666 MHz	98 GB @ 2400 MHz
Instruction set architecture	AVX2	AVX-512F,DQ,CD,BW,VL	AVX-512F,PF,ER,CD
Theoretical peak flops (FP32 / FP64)	2649 / 1324	6144 / 3072	6092 / 3046

4.2 GSMNP LiDAR Benchmark Problem

- For a benchmark problem, we cluster a data set from Kumar et al. 2015 (`https://doi.org/10.3334/ORNLDAAC/1286`): airborne multiple return Light Detection and Ranging (LiDAR) surveys of the Great Smoky Mountains National Park (GSMNP).
- LiDAR enables large scale remote sensing of topography, built infrastructure, and vegetation structure.
- k-means clustering of LiDAR point cloud data was used to construct vertical density profiles to characterize vertical vegetation structure.
- 30 m $\times$ 30 m horizontal and 1 m vertical (extending to a height of 75 m) spatial resolution was used, with the input data set consisting of 3,186,679 observations, each of 74 variables, requiring 900 MB of storage in single precision.

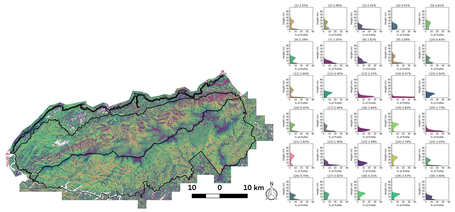


Figure 4: *Vegetation structure classes and their distribution in the Great Smoky Mountains National Park (GSMNP) derived from k-means clustering ($k = 30$) of the GSMNP LiDAR data set. The spatial distribution is shown at the left, and the prototype canopy structures (cluster centroids) are shown at the right. The color scheme on the map correspond to the colors of the prototypes.*

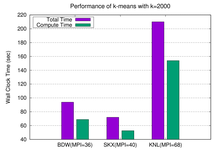


Figure 5: *Baseline performance (employing a version of pKluster that incorporates the triangle inequality-based "acceleration" algorithm but no further code improvements) of the three benchmarking platforms for computing $k = 2000$ clusters for the GSMNP LiDAR data set using the accelerated k-means algorithm.*

4.3 Effective Use of Hyperthreads

- Using a pure MPI approach (one MPI rank per core), performance of the accelerated k-means clustering approach is surprisingly poor on the "Knights Landing" (KNL) processor.
- Using two MPI ranks per core slightly decreases time in the actual clustering calculation, but slightly increases total time due to greater overhead in master-worker coordination.
- This suggests that using more available hardware threads can improve performance on KNL, if we can avoid increasing master-worker overhead.
- We introduced OpenMP threading in the most time-intensive routine, `cluster_aliquot()`, using dynamic scheduling to deal with inherent load imbalance in the accelerated approach. This provides the most benefit on KNL, but BDW and SKX also see a performance boost.

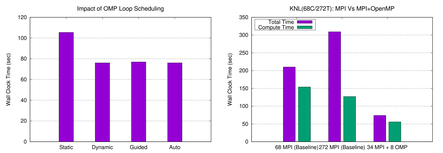


Figure 6: *Impact of incorporating OpenMP threading into pKluster on the KNL platform. At left: The effects of different kinds of OpenMP loop scheduling when finding $k = 2000$ clusters of the GSMNP data set. Setting* `schedule` *to* `dynamic`, `guided`, *or* `auto` *provides significant speedup over the* `static` *default. At right: Performance for various MPI rank and OpenMP thread configurations. CLUSTER_TIMER denotes the time spent inside the actual k-means calculation (no communication or I/O), and TOTAL_TIMER denotes the entire wall-clock time for completing the pKluster execution. Using all 272 hyperthreads significantly benefits performance when using the combination of 34 MPI ranks and 8 OpenMP threads per rank.*

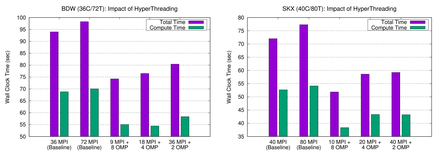


Figure 7: *Comparison of times to cluster the GSMNP LiDAR data set with $k = 2000$ on the Broadwell (BDW, left) and Skylake (SKX, right) Xeon processors for different numbers of MPI ranks and OpenMP threads.*

4.4 Improving Computational Intensity Using Level-2/3 BLAS

- We recently realized that it is possible to achieve greater computational intensity of the observation–centroid distance calculations by expressing the calculation in matrix form:
  - For observation vector x_i and centroid vector z_j, the squared distance between them is $D_{ij} = \|x_i - z_j\|^2$.
  - Via binomial expansion, $D_{ij} = \|x_i\|^2 + \|z_j\|^2 - 2x_i \cdot z_j$
  - The matrix of squared distances can thus be expressed as $D = \overline{x}\mathbf{1}^\intercal + \mathbf{1}\overline{z}^\intercal - 2X^\intercal Z$, where X and Z are matrices of observations and centroids, respectively, stored in columns, $\overline{x}$ and $\overline{z}$ are vectors of the sum of squares of the columns of X and Z, and 1 is a vector of all 1s.
- The above expression for D can be calculated in terms of a level-3 BLAS operation (xGEMM), followed by two rank-one updates (xGER, a level-2 operation).
- Level 2 and 3 BLAS operations admit very computationally efficient implementations; we use the highly optimized BLAS implementations from Intel's MKL and NVIDIA cuBLAS Xeon Phi and GPGPUs, respectively.
- We have found that the above, matrix formulation for the distance calculations is dramatically faster than the straightforward loop over vector distance calculations when many distance comparisons must be made.
- Using the matrix formulation for distance comparisons in early k-means iterations is straightforward; a more complicated approach we will explore is using the matrix formulation in combination with the acceleration techniques described above, in which only a subset of observation–centroid distances are calculated.

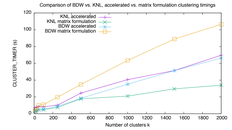


Figure 8: *Comparison of timings for clustering the GSMNP LiDAR dataset for different values of k on the KNL and BDW platforms using the accelerated k-means algorithm and the matrix formulation that uses level-2 and level-3 (xGEMM) BLAS calls.*

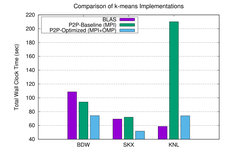


Figure 9: *Performance comparison of k-means implementations for the GSMNP LiDAR dataset with $k = 2000$. Here P2P refers to the pKluster implementation using triangle inequality-based acceleration, and BLAS refers to the matrix formulation of distance computations.*

5. Additional Geospatio(temporal) Applications

5.1 MODIS-based Phenoregionalization and Change Detection

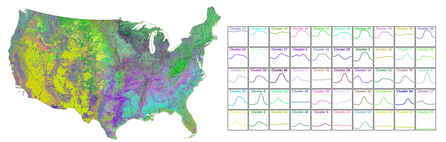


Figure 10: *At left: a map of "phenoregion" assignments for the year 2012, based on k-means analysis with $k = 50$ of the entire MODIS-derived ForWarn NDVI product for years 2000–2012. The body of observation vectors being clustered consists of the year-long MODIS NDVI time series for every map pixel, for each year. The map indicates cluster membership (in random colors) for the phenology observed in 2012 at each map pixel. At right: The fifty centroids (corresponding to "phenoregion" prototypes) used for the membership assignments in the map.*

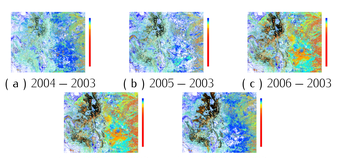


Figure 11: *Maps showing the relative state space transition distances (how different phenoregion assignments are for given years) between years in Colorado and southern Wyoming. Pine beetle mortality correlates strongly with high transition distances. Black-outlined polygons are disturbed areas indicated on aerial sketch maps.*

5.2 Analysis of Global Climate Regimes

- We can compute climate-based "ecoregions" using k-means analysis of bioclimatic plus ancillary variables.
- Comparing the maps produced for present day and for simulated future conditions facilitates quantitative study of the effects of projected climate change on ecoregion distribution.

Table 2: *Variables used for delineation of global climate regimes. Data drawn from Hijmans et al. 2005 [doi:10.1002/joc.1276], Saxon et al. 2005 [doi:10.1111/j.1461-0248.2004.00694.x], Baker et al. 2010 [10.1007/s10584-009-9622-2]*

Variable Description	Units
Bioclimatic Variables	
Precipitation during the hottest quarter	mm
Precipitation during the coldest quarter	mm
Precipitation during the driest quarter	mm
Precipitation during the wettest quarter	mm
Ratio of precipitation to potential evapotranspiration	-
Temperature during the coldest quarter	°C
Temperature during the hottest quarter	°C
Day/night diurnal temperature difference	°C
Sum of monthly T_{mn} where $T_{mn} \geq 5$°C	°C
Integer number of consecutive months where $T_{mn} \geq 5$°C	–

Variable Description	Units
Edaphic Variables	
Available water holding capacity of soil	mm
Bulk density of soil	g/cm^3
Carbon content of soil	g/cm^2
Nitrogen content of soil	g/cm^2
Topographic Variables	
Compound topographic index (relative wetness)	–
Solar interception	(kW/m^2)
Elevation	m

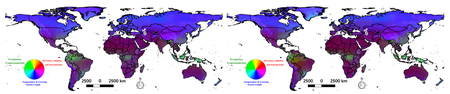


Figure 12: *1000 Global climate regimes generated by the k-means clustering algorithm for contemporary time period (left) and predicted future 2100 by HadCM3 climate model under A1FI emissions scenario (right). Clusters are colored according to a* similarity color *scheme using the top three components from principal components analysis. The red color channel largely reflects topography and soil properties; the green channel, precipitation variables and evapotranspiration; and the blue channel, temperature variables and growing season length.*

6. Future Directions

6.1 pKluster Software Development

- Investigate hybrid approach combining accelerated k-means method and matrix formulation within the same iteration.
- Re-implement a fully distributed, masterless approach in the current version of the code to handle cases in which master-slave overhead is high (e.g., many cases on KNL).
- Add support for emerging high-capacity, non-volatile memory technologies.
- Supported open-source release under Apache License 2.0.
- Also considering a new implementation using PETSc, the Portable, Extensible Toolkit for Scientific Computation (`https://www.mcs.anl.gov/petsc/`).

6.2 Future Science Goals

- Increasingly powerful CPUs, combined with large byte-addressable memories promised by emerging nonvolatile random access memories (NVRAM) technologies, will allow more ambitious analysis of large geospatiotemporal data sets from both models and empirical observations.
- Potential questions of interest:
  - How is climate change affecting global plant distributions?
  - What are the implications for global carbon budgets?
  - What changes do we expect to key events like onset of growing season?
  - What changes do we expect to suitable growing ranges for crops?
  - Are there policy implications for agriculture and ensuring the food supply?
- Could combine analysis all of the MODIS vegetative phenology record with global fine-scale meteorological reanalysis (gridded reconstruction of meteorological history) and possibly other ancillary data layers.
  - Enables attribution of vegetation changes to climate or other events.
  - Study directly observed vegetation responses to meteorology.
- Could analyze high-resolution and/or multi-model ensemble Earth system model simulations:
  - Project changes to distribution of eco-phenoregions (identified by the historical analysis) for different climate change scenarios.
  - Combine with crop physiology models to project changes in yields.
  - Combine with urban growth models, population models to assess resource planning, policy scenarios, crop futures.

7. Acknowledgments

R. T. Mills was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. JK and FMH were partially supported by the Next Generation Ecosystem Experiments - Arctic (NGEE Arctic) project, which is sponsored by the Terrestrial Ecosystem Sciences (TES) Program, and the Reducing Uncertainties in Biogeochemical Interactions through Synthesis and Computation Scientific Focus Area (RUBISCO SFA), which is sponsored by the Regional and Global Model Analysis (RGMA) Program. The TES and RGMA Programs are activities of the Climate and Environmental Sciences Division (CESD) of the Office of Biological and Environmental Research (BER) in the U.S. Department of Energy Office of Science. WWH, JK, and FMH claim additional support from the Eastern Forest Environmental Threat Assessment Center (EFETAC) in the U.S. Department of Agriculture Forest Service. This presentation has been coauthored by UChicago Argonne, LLC and UT-Battelle, LLC under Contract numbers DE-AC02-06CH11357 and DE-AC05-00OR22725, respectively, with the U.S. Department of Energy.