Phylanx Year 3 Kickoff Meeting

November 7, 2019

- Introductions
  - We will be giving sort talks of the work we are doing
  - Adrian
    - I would like us over the course of this meeting to solidify the product we want to produce
    - It needs to satisfy two criteria:
      - Meet the requirements of our sponsor
      - Be able to secure follow-on funding
- Hartmut- Overview of Phylanx
  - Parallelism: We decided to go with a strict SPMD parallelism
  - Status:
    - Frontend
      - Python Bindings
      - Keras Backend (80% done)
      - Jupyter Integration
      - Traveler Tools
    - Middleware
      - Theoretical progress
      - Starting to perform experiments
    - Backend
      - 150 primitives
  - Outlook
    - Distributed processing
    - Front-end
      - NumPy interface
      - Higher level frontend
    - Tooling support
      - Traveler tools
  - To accomplish in this meeting:
    - Define Minimal Viable Product
      - Satisfy sponsor
      - Enable us to secure funding
- Avah – Matrix Tiling
  - I have assumed that we have dense, large, and tiled matrices
    - Tiles are uniform
    - Each matrix is write-once
  - We need to be careful to use optimal memory techniques as it will quickly become easier to just to keep two copies of the matrix, tiled both ways
  - Looking Forward
    - Determination of the *Hp* given a user program
      - May approximate by running the program a few times
    - Experimentally solving MTP on *Hp*
      - We should avoid over-engineering

    - Guoli and I are working on Online Algorithms (Online Mincut, and other sub-modular functions) and the k-server problems on trees and circles
    - Takeaway- we need to start working on mining information from the program
        - We need to focus on implementation over the next year
        - Chris- We may need a feed-back mechanism to let the user know more optimal ways of executing the AST
- Maxwell – Distributed Primitives
    - We have
        - Map operations
        - Distributed data structures
            - distributed_object, distributed_vector, distributed_matrix
        - Distributed Primitives
            - Dot_d 2d2d
            - Cannon's Algorithm
            - Distributed CSV Read
                - I hope to finish this soon
- Chris – Sponsor Focus
    - My bosses are interested these two things
        - Distributed Memory
        - Visualization Tooling
            - APEX
            - Traveler-Integrated
            - K8 integration
    - Feed-back I have received
        - There are a lot of developers which are not looking at what they are doing
            - Spark -> Has IO issues, popular, can **map unstructured data**
                - Perhaps it is just Pandas
        - Government has been outsourcing IT to Amazon and Azure
            - Kubernettes integration
                - Sylabs is working on wlm-operator
                    - Cloud can talk to HPC
                    - "serverless" platforms
            - Steve- this sounds like a Science Gateway
                - We would use tech like Agave or Open Science Gateway
            - I would like to have the **Kubernettes integration by next summer**
    - Future
        - Resiliency/Fault Tolerance/Debugging
        - I/O story (HDF5? HPX-IO? ADIOS-2?)
        - Tiling with space filling curves (improve cache)
- Kevin – APEX
    - We need a global view of data collection
    - Want to do critical path analysis
    - Hartmut- I would like to discuss how APEX and HPX are producing data
- Kate
    - Current work
        - Atria
        - Atria – Jupyter Version

        - Roundtrip
        - Gantt charts
        - Traveler Integrated
    - What next?
        - Move prototypes into Traveler Integrated
        - Accessibility
        - Visualize tiling?
        - Scale?
            - Keep things responsive
    - How do we build all of this
        - Build system integration?
        - Docker Files
        - Spack
- Steve
    - Working Docker Environment
    - Teaching Support
        - JupyterHub
        - Cling
        - @Py11 decorator
        - Refactored the help function
        - Distributed Garbage Collection
            - HPX Component Maker
- Bita – Keras Backend
    - Keras multi-backend support will be ending soon
        - Should we continue to work in this direction
        - Rod- We could use this work to support TensorFlow
            - Hartmut- Let's look at your examples
        - Chris- Either backend is fine **I just want a training piece added**
- Shahrzad – HPX Backend for Blaze
    - Effect of grainsize on execution time
    - Universal Scalability Law
    - Next Step
        - Find the range of the flat region of grain size
        - Make things fit into cache (choose a small block_size while number of columns is divisible by cache line
        - Find the range of chunk_sizes for the given range of grain size
        - Generalize model to integrate the matrix size
- Bibek – Task Inlining
    - Lambda min
    - Latest comparisons of Python and Phylanx
    - Chris – This research that [everyone] is doing is what I was hoping to achieve with this project
- Tianyi – hpxMP
    - Performance of hpxMP versus LLVM and GOMP
- Nanmiao – Blaze Iterative Solvers
    - Linear System Solvers
    - Previous Work

        - Conjugate Gradient
        - Bi Conjugate Gradient Stabilized
        - Preconditioned BICGSTAB
    - Added four solvers
        - Preconditioned CG
        - GMRES (Generalized minimal residual method)
        - Lanczos
        - Arnoldi
- Rod – Asynchronous Execution of Distributed Python
    - Phylanx
        - Targets distributed and parallel HPC
        - Distributed Programming in Python
        - Asynchronous Execution
        - Optimization
        - Highlights
            - Distributed Algorithms
            - Higher-dimension NumPy (Blaze Tensor)
            - Linear Algebra on GPUs (cuBlaze)
            - Optimization of linear algebra operations
            - CNN (Convolutional Neural Networks) [i.e. Keras]
            - Scheduling Algorithms
            - hpxMP
            - Solvers for peridynamic simulations
            - User Front-ends
                - Docker
                - Jupyter Notebooks
        - Ideas
            - Expose PhySL to Python
            - Canonize the Program
            - Create the task graph
            - Tile the Data
            - Optimize the task graph
            - Debugging
        - I think PhySL is the key here
            - It transforms a declarative program into a graph
        - We need to build
            - Generic Python to PySL code
            - Provide modules for widely used solutions (Keras, NumPy, TF)
            - Great Visualizations for perf. analysis and debugging
- Phylanx Demo
    - Adrian- I suggest we define the product we want to produce by the end of this project
    - Operation Jetlag
        - Audience: Domain Scientists
            - Array processing
            - “Data Analysis”
            - ML/DL
        - Problems to Demo:

                - Contracted Algorithms
                    - Some in distributed
                - Keras code using Phylanx backend (training example)
                - Solve a performance problem with Traveler Integrated
            - Platform: Jupyter Notebook w/ HPC Backend
                - Deploy backend in Docker Container
                - Use Agave to manage cluster deployment
                - Returns result and link to Traveler visualization
            - Other Key Aspects to Demonstrate
                - Task dependencies for a distributed application
                - Interactivity between Traveler Tree, Gantt Chart, etc.
                - Demonstrate Traveler solving a performance problem locally before launching the application remotely
                - Performance Comparisons?
    - Requires:
        - LSU
            - Implement remaining algorithms
            - Complete Keras implementation
            - Enhance distributed Phylanx
            - Add features to Jupyter Notebook
                - Deploy button
                - Add link to Traveler visualization
                - Integrate with Agave
        - UA
            - Work on interactivity in Traveler Integrated
            - Add performance comparison ability?
        - UO
            - Enable dependency tracing across a cluster
    - We need to start “testing” this demo ASAP
        - Use this to drive development
- Elevator Pitch for Phylanx
    - Some key words to choose from:
        - Productive
        - Conformant
        - Supported (Tooling)
        - Performant
        - Composable
- Closing
    - We will meet tomorrow at 9am
    - Will plan breakout sessions tomorrow
    - Kevin
        - Meet with Patrick and Steve about SIAM proposal
        - Meet with Hartmut about APEX – HPX interactions
            - Buildsystem
            - Performance Counter duplication
            - Requiring APEX testing in HPX builders
        - Meet with Kate