Colmena: Seamless Computational Campaigns across Multiple Computing Clusters with Parsl/FuncX and Object Proxies

Logan Ward
Assistant Computational Scientist
Data Science and Learning Division
Argonne National Laboratory

14 September 2022
Strategies for steering computational campaigns are complicated

*Parallel Optimizers: A “simple” example with no optimal strategy*

**Better system utilization**
**Fewer calls to “select next tasks”**

**Batch Optimizer**
- Wait for N tasks to complete, then pick next batch

**Streaming Optimizer**
- Pick new tasks as soon as one completes

**Interleaved Optimizer**
- Maintain a task queue

Colmena provides simplifies expressing steering strategies
Colmena is a wrapper over Exascale Workflow tools

**Programming Model: Task Queues**

# Primitive Units

```python
queue.send_inputs(1)
result = queue.get_result()
```

**Programming Model: Agents**

```python
class Thinker(BaseThinker):
    @agent
def make_work(self):
        self.queue.send_inputs(1)
```

**Task Server:**
- Dispatches work requests to compute
- Communicates results back to thinker

**Backend:**
- Supports most HPC and cloud services
- Easily configure multiple worker types, multi-site workflows
- Limited support for ensembles of MPI applications
- *Future:* Balsam, FuncX, RCT
Example application: “Interleaved,” AI-in-the-loop optimizer

Retasking nodes between jobs…

…yields more science per compute-hour.

Details: Ward et al. ML4HPC, SC21.
So, what’s new in ‘22?

*Multi-site Campaigns!*
Why multi-site? *Moving compute onto best hardware*

- Most of our work is CPU-only (needs: many, cheap)
- The “retrain” tasks are slow on CPU (needs: specialized)
- Faster retraining means better steering
How multi-site? *FuncX*

It’s just *FuncX*. We use the “FuncXExecutor,” so it acts like Parsl
How good multi-site? *Same performance, less port headache*

Machine learning tasks take only 3% longer than best-effort with SSH tunnels

Scientific outcomes are identical

(a) ML Time-to-Solution (s)

(b) Molecules Found

- **Parsl**
- **Parsl+Redis**
- **FuncX+Globus**

Node Hours Expended (hr)
Colmena lets you explore computational cost tradeoffs

Steering policies tradeoff between time to solution, GPU time, and CPU time.
Conclusions and Future for Colmena

What did we cover today?

• Colmena lets you build complex steering policies

What to watch for next year?

• This work published (at least on ArXiV!)

• A perspective on ensemble steering toolkits
  – “How are libE and

• More Colmena applications
  – Fitting machine-learned surrogates for simulations
  – Coordinating simulation self-driving laboratories
  – Rapid screening of HPC

• Integration with more workflow engines (e.g., RCT!)
Got opinions about what Colmena is? Join our interest group!

Email me (lward@anl.gov) if you want to join the conversation!
# Acknowledgements: The (growing!) team

<table>
<thead>
<tr>
<th><strong>Argonne: ExaLearn</strong> – Using AI with HPC</th>
<th><strong>Argonne: JCESR</strong> – Molecular modeling for batteries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yadu Babuji, Ben Blaiszik, Ryan Chard, Kyle Chard, Ian Foster, Greg Pauloski, Ganesh Sivaraman, Rajeev Thakur</td>
<td>Rajeev Assary, Larry Curtiss, Naveen Dandu, Paul Redfern</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>MoISSI</strong> – Workflows for quantum chemistry</th>
<th><strong>PNNL: ExaLearn</strong> – Graph algorithms for learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lori A. Burns, Daniel Smith, Matt Welborn, <em>many other open-source contributors</em></td>
<td>Sutanay Choudhury, Jenna Pope</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>BNL: ExaLearn</strong> – Optimal experimental design</th>
<th><strong>Argonne ALCF</strong> – AI, Data and Simulation on HPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frank Alexander, Shantenu Jha, Kris Reyes, Li Tan, Byung Jun, <em>and more</em></td>
<td>Murali Emani, Alvaro Vazquez-Mayagoitia, Venkat Vishnawath</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>FuncX</strong> – Seamless multisite deployment</th>
<th><strong>ExaWorks</strong> – Interfacing to HPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kevin Hunter Kesling, Kyle Chard, Ryan Chard, Ben Clifford, <em>and more</em></td>
<td>Ayman Alsaadi, Matteo Turilli, Shantenu Jha, Kyle Chard</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Ensemble Group</strong> – Defining ensemble needs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>John-Luke Navarro, Jonathon Ozik, Tom Peterka, Stephen Hudson, Orçun Yildiz, Alex Brace, Arvind Ramanathan, <em>and more</em></td>
<td></td>
</tr>
</tbody>
</table>