

DHPVM:

A Distributed Heterogeneous Parallel Virtual Machine

Dan Plyukhin and Husnain Mubarik

CS 598 LCE, Spring 2021
UIUC

Motivation

- **Mobile battery-powered devices** (earbuds, smart watches, VR, drones, ...)
 - Hard requirements: *performance* and *energy efficiency*
 - Soft requirements: *programmability* and *portability*
- **(Approx)HPVM:**
 - Write *one* program to target *many* heterogeneously parallel architectures
 - Offload computation to onboard coprocessors
- **DHPVM:**
 - What about **remote coprocessors** on nearby edge hardware?
 - 5G base stations; open laptops; public computing hubs?
 - Earable Computing
 - Real time conversational agents?
 - Real time language translation / NLP (ALBERT, GoogleBERT)?
 - Real time navigation?
 - Availability of remote coprocessors constantly in flux
 - Each edge device can have a different architecture!
 - Need an IR; can't compile the program in advance

Related Work

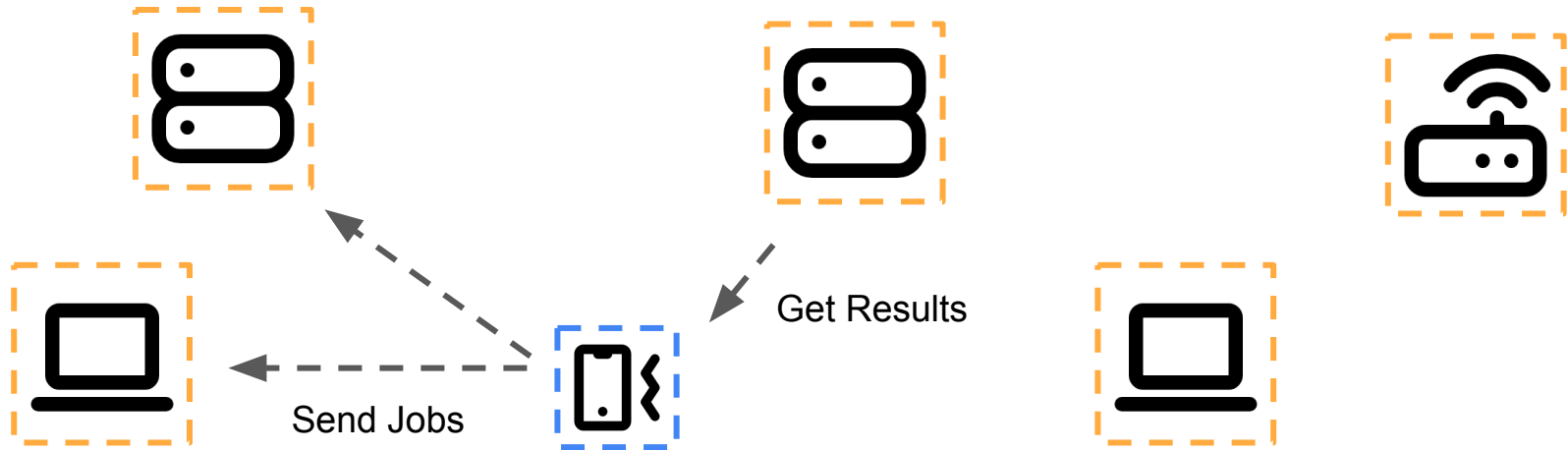
- **Remote code offloading**
 - VM-based offloading: extremely coarse-grained
 - MAUI uses method-level offloading; uses CLR for portability
 - SIMDOM: Dynamic binary translation with support for vector instructions
- **Application placement**
 - e.g. Hadoop, Spark, mCloud, Wishbone...
 - How do we decide which job to offload to whom?
 - Many creative solutions - but there's no single right answer
- **Macroprogramming / multi-tier programming**
 - e.g. TinyDB, Pleiades, Kairos, Chor, ...
 - “Local view”: Program each thread and each process independently
 - “Global view”: Write a single program that *compiles into* per-process programs

Goal: Distributed Architecture

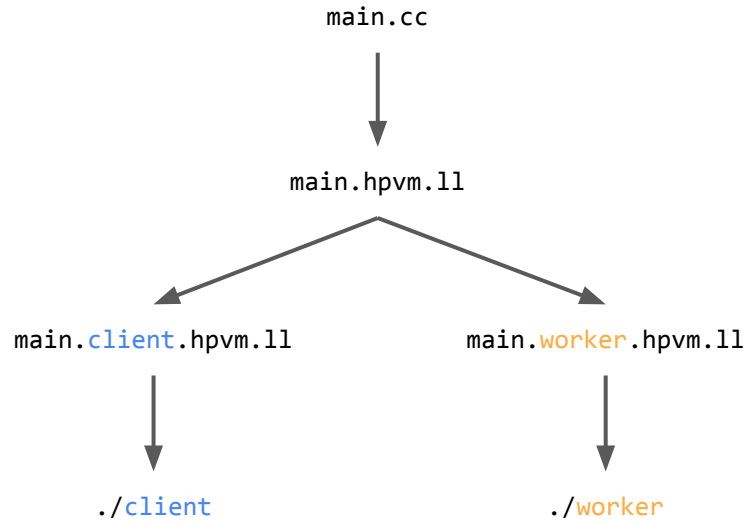
A battery-powered mobile device (the **client**) connects to nearby edge hardware (the **workers**)

The DHPVM runtime sends **jobs** to workers and waits for a response

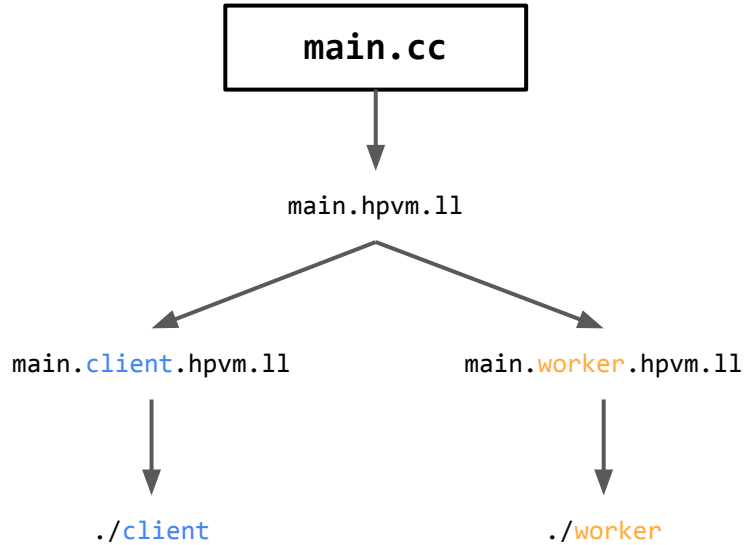
Jobs should be pure functions to support mobility and fault-tolerance



Compilation Process



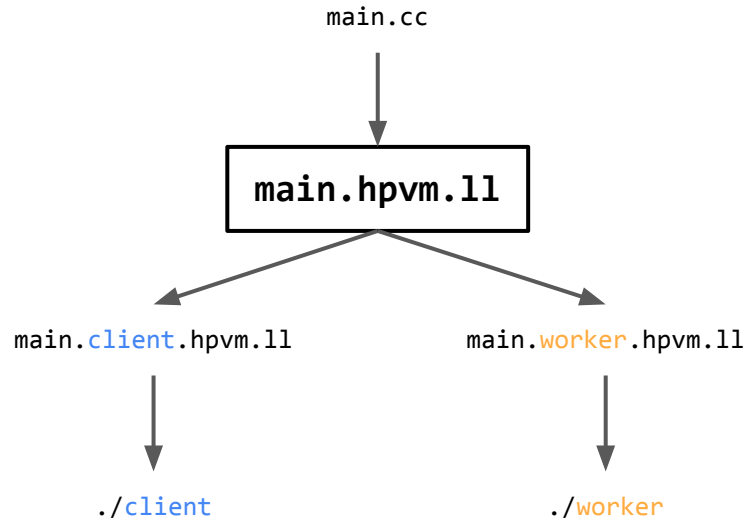
Compilation Process



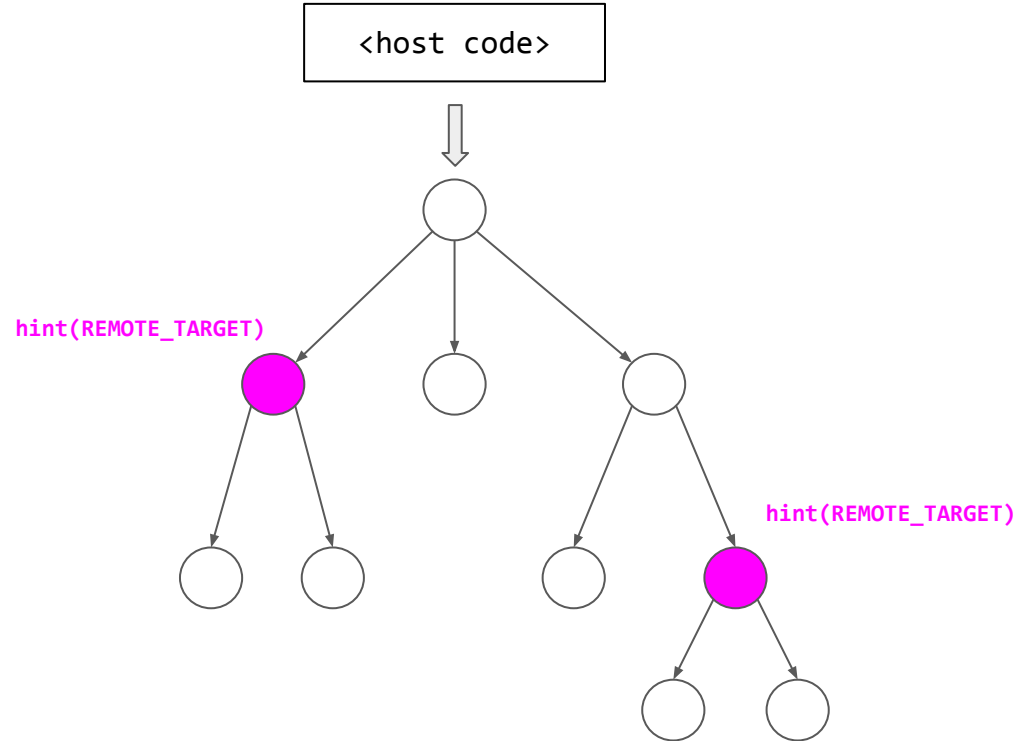
HPVM-C File with REMOTE_TARGET Hints

```
...  
void internal(...) {  
    __hpvm__hint(REMOTE_TARGET);  
    __hpvm__createNodeND(..., leaf, ...);  
    ...  
}  
  
void root(...) {  
    __hpvm__hint(CPU_TARGET);  
    __hpvm__createNodeND(..., internal, ...);  
    ...  
}  
  
int main() {  
    ...  
    __hpvm__launch(root, args);  
    ...  
}
```

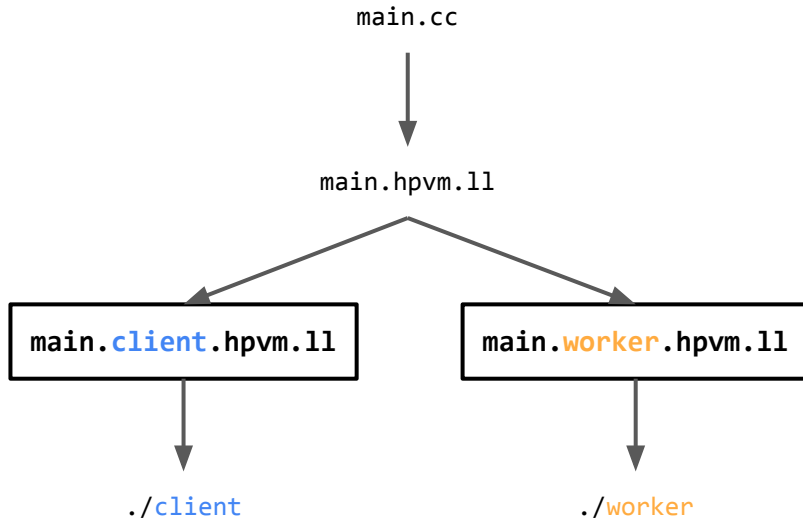
Compilation Process



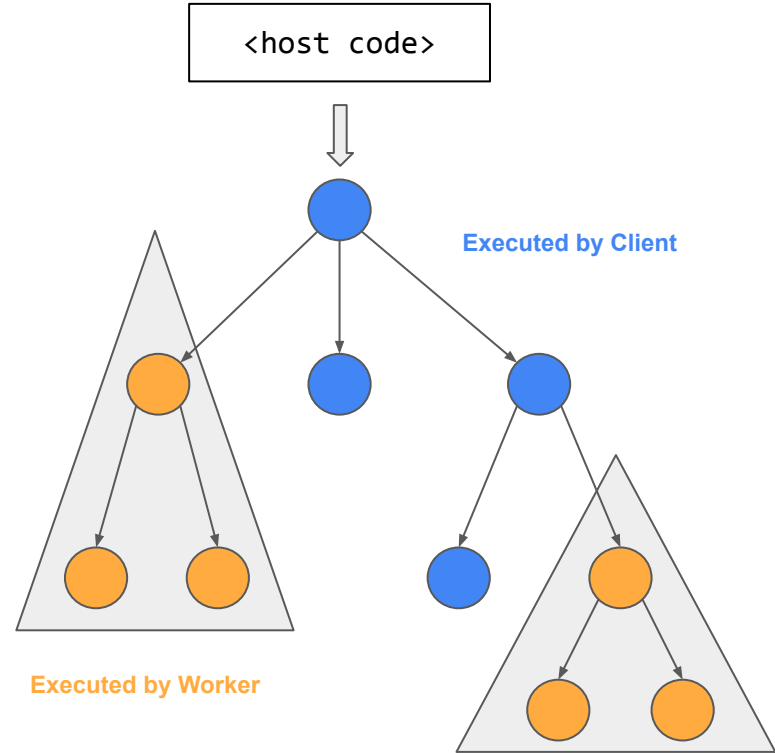
LLVM Bitcode with Hierarchical HPVM DFG



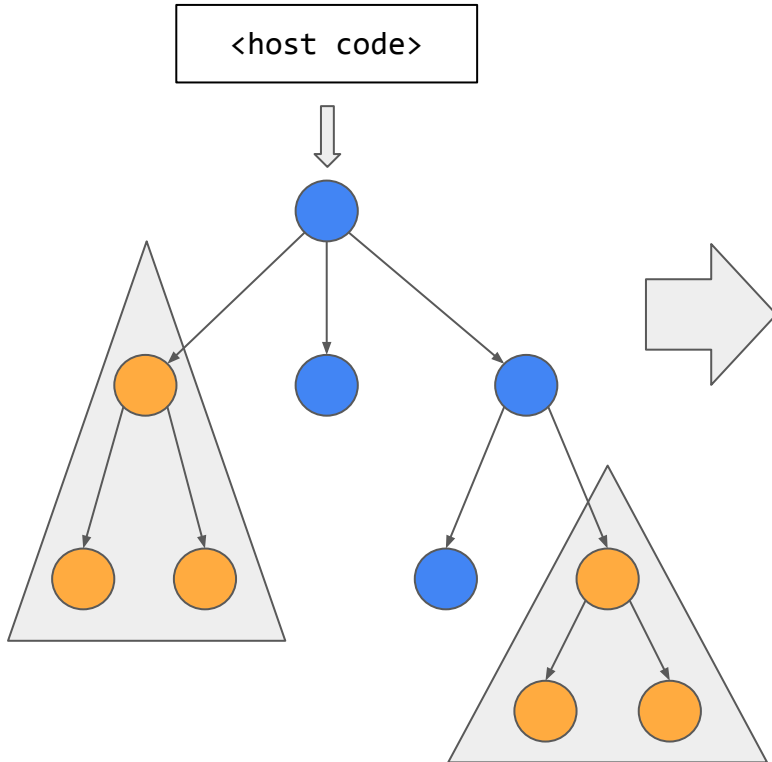
Compilation Process



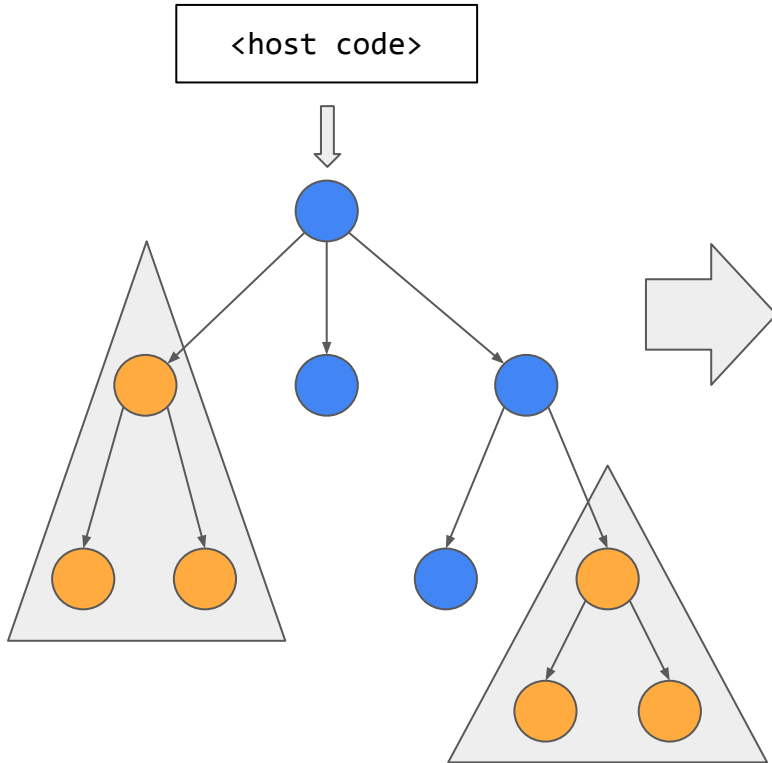
LLVM Bitcode with Hierarchical HPVM DFG



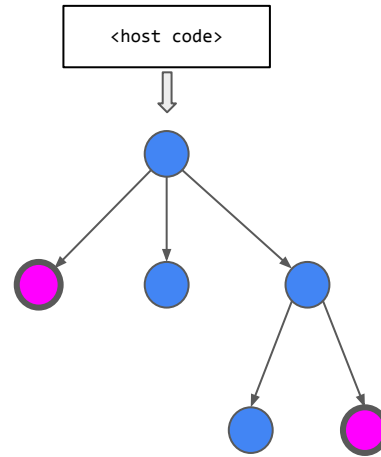
Compilation Process



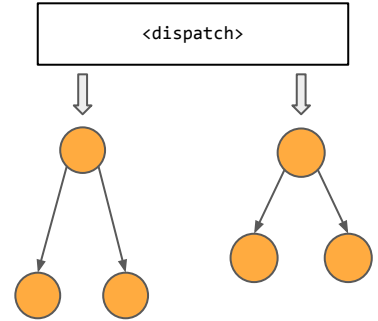
Compilation Process



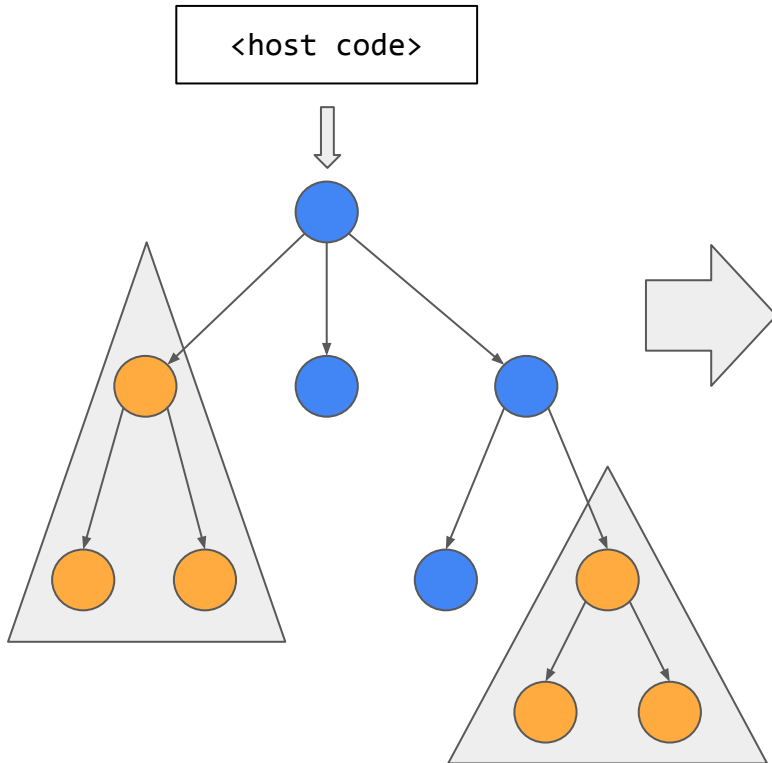
main.client.hpvm.11



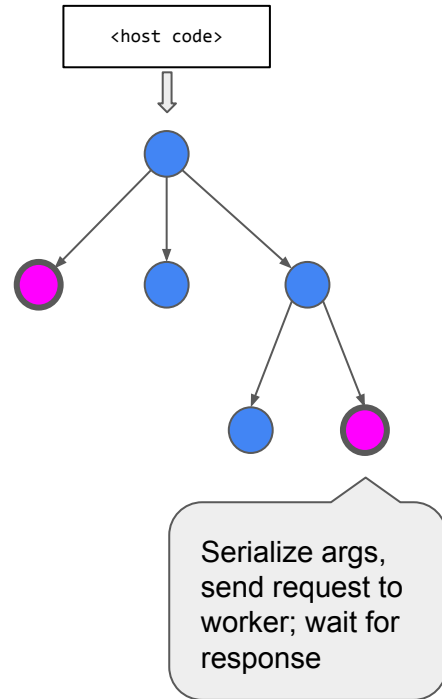
main.worker.hpvm.11



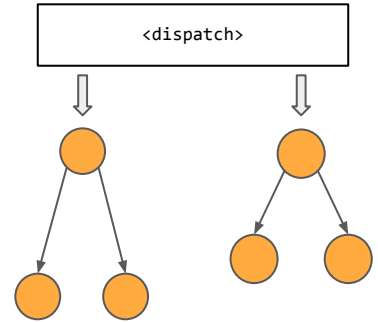
Compilation Process



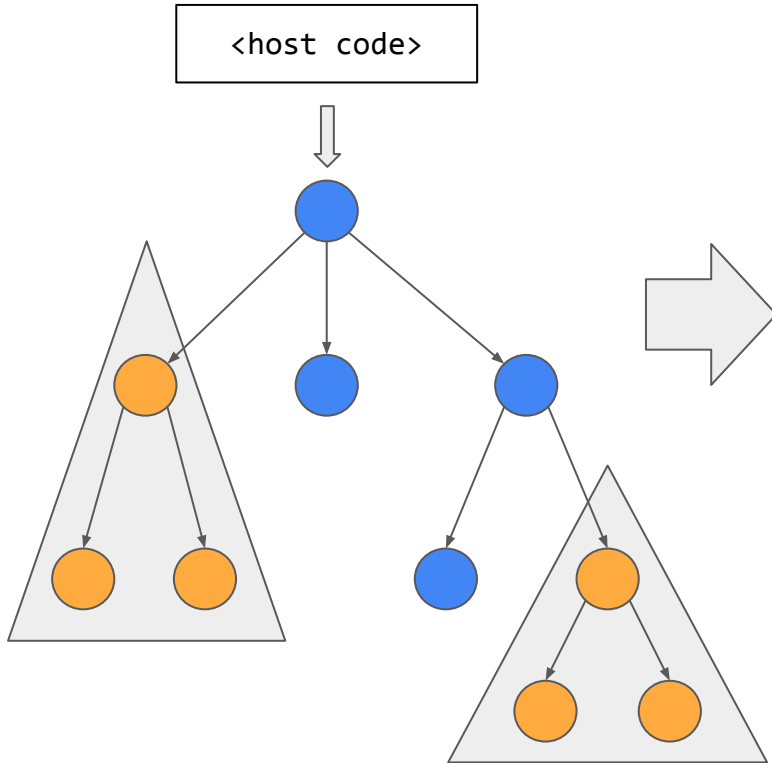
main.client.hpvm.ll



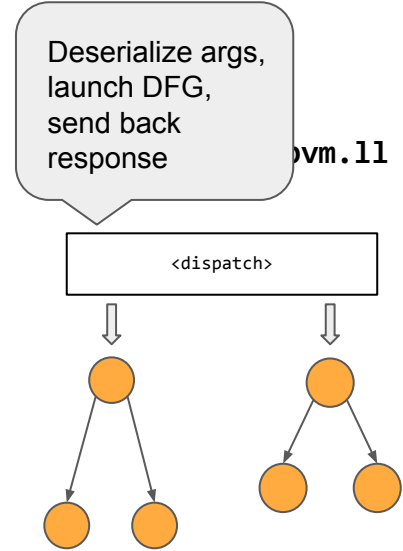
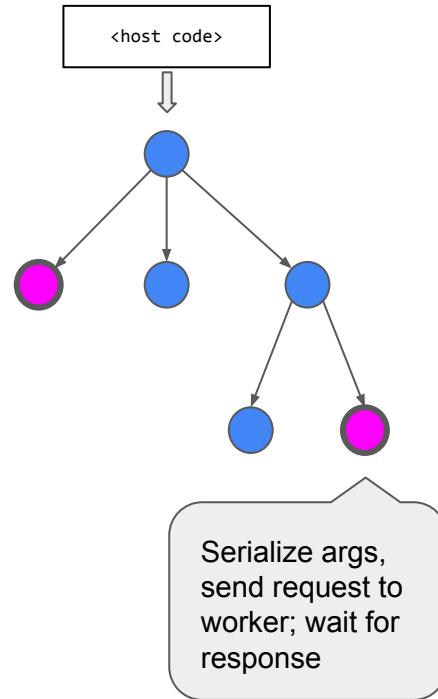
main.worker.hpvm.ll



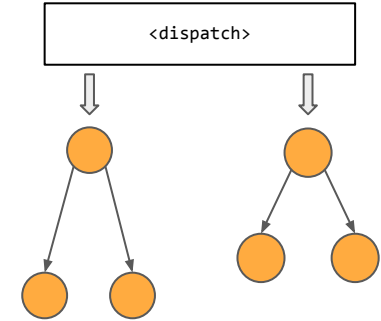
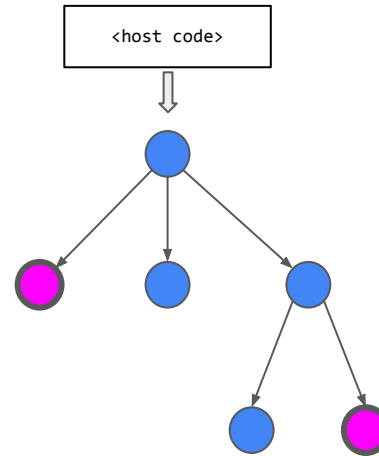
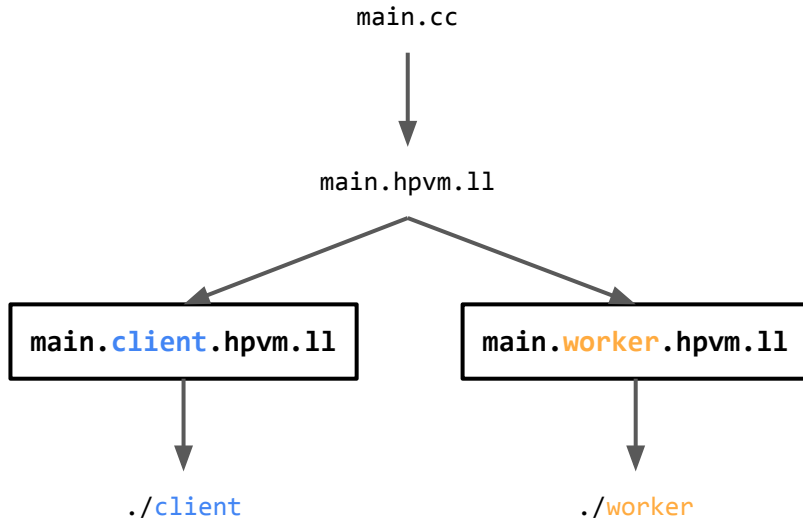
Compilation Process



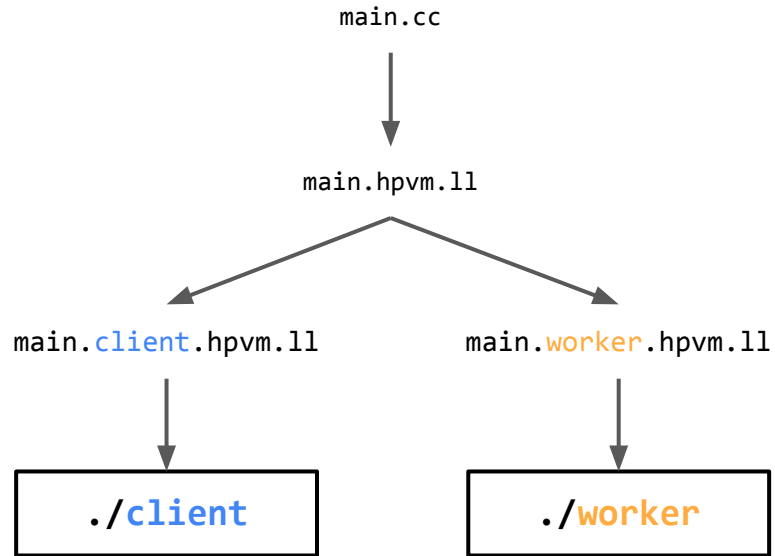
main.client.hpvm.11



Compilation Process



Compilation Process



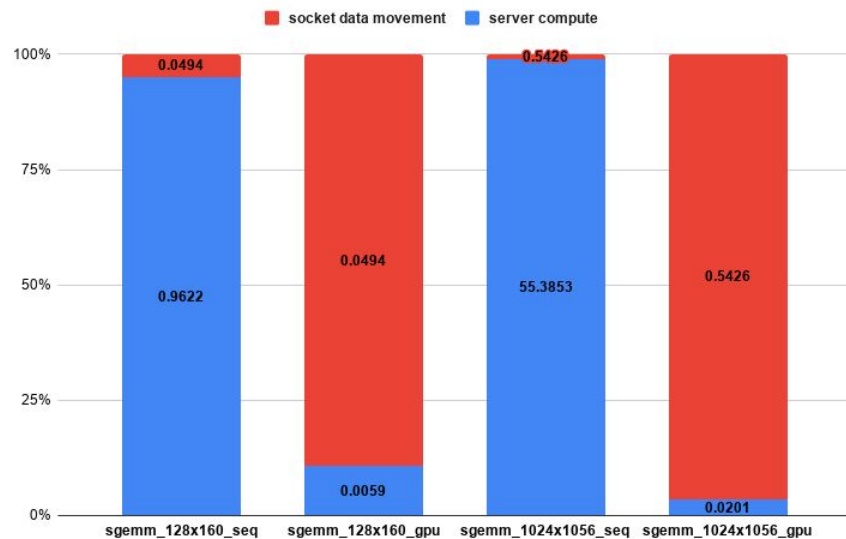
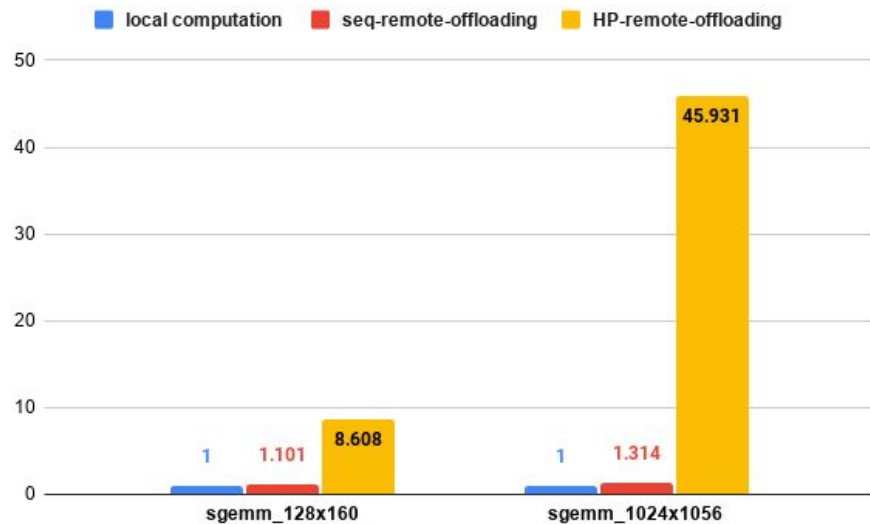
<client exe>

<worker exe>

Runtime

- Evaluation HW
 - a. Local Machine: i5-3470 (No GPU)
 - b. Remote Server: E5-2680 (2x Titan Xp)
- Benchmark:
 - a. SGEMM from Parboil benchmark suite for two matrix sizes (128x160, 1024x1056)
- Local Computation: Execution on local machine (compiled for TARGET=seq)
- Seq-Remote-Offloading: Distributed computing (compiled for TARGET=seq)
 - Client executable running on local machine
 - Worker executable running on server machine
 - ZMQ socket connection on port 5555 connected via ssh
- HP-Remote-Offloading: Remote code offloading (compiled for TARGET=GPU)
 - True DHPVM, client, worker, zmq socket, ssh

Evaluation



Conclusion

- DHPVM is potentially another “killer app” for the HPVM IR
- Our evaluation shows that:
 - Seq-remote-offloading: 10% to 31% performance improvements
 - HP-remote-offloading: 8.6x to 45.9x performance improvements
 - CPU target is bottlenecked by compute cost
 - GPU target is bottlenecked by data movement cost
 - Overall data movement cost is low 0.04 - 0.5 seconds for small and medium matrix respectively.

Future work:

- Support for streaming computation
- Concurrent offloading to multiple workers
- Customizable offloading policies
 - Deciding which DFG nodes should be offloaded
 - Deciding which worker should receive the work
- Shipping HPVM bitcode at runtime
- Distributed Earables Project (hopefully ASPLOS!)

Thank
You!

Bibliography (part 1)

- Aazam, Mohammad, Sherali Zeadally, and Khaled A. Harras. "Offloading in Fog Computing for IoT: Review, Enabling Technologies, and Research Opportunities." (2018)
- Alam, Md. Golam Rabiul, Yan Kyaw Tun, and Choong Seon Hong. "Multi-Agent and Reinforcement Learning Based Code Offloading in Mobile Fog." (2016)
- Author, Anonymous. "Jarvis: An Adaptive System for Scalable Datacenter Monitoring and Analytics." (2021)
- Azzarà, A., D. Alessandrelli, S. Bocchino, M. Petracca, and P. Pagano. "PyoT, a Macroprogramming Framework for the Internet of Things." (2014)
- BenSaleh, Mohammed Sulaiman, Raoudha Saida, Yessine Hadj Kacem, and Mohamed Abid. "Wireless Sensor Network Design Methodologies: A Survey." (2020)
- Brogi, Antonio, Stefano Forti, Carlos Guerrero, and Isaac Lera. "How to Place Your Apps in the Fog: State of the Art and Open Challenges." (2020)
- Carbone, Marco, and Fabrizio Montesi. "Deadlock-Freedom-by-Design: Multiparty Asynchronous Global Programming." (2013)
- Cardellini, Valeria, Vincenzo Grassi, Francesco Lo Presti, and Matteo Nardelli. "Optimal Operator Placement for Distributed Stream Processing Applications." (2016)
- Cuervo, Eduardo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. "MAUI: Making Smartphones Last Longer with Code Offload." (2010)
- Culler, D. E., A. Dusseau, S. C. Goldstein, A. Krishnamurthy, S. Lumetta, T. von Eicken, and K. Yelick. "Parallel Programming in Split-C." (1993)
- Dias de Assunção, Marcos, Alexandre da Silva Veith, and Rajkumar Buyya. "Distributed Data Stream Processing and Edge Computing: A Survey on Resource Elasticity and Future Directions." (2018)
- Dulay, N., M. Micheletti, L. Mostarda, and A. Piermarteri. "PICO-MP: De-Centralised Macro-Programming for Wireless Sensor and Actuator Networks." (2018)
- Flinn, Jason, and M. Satyanarayanan. "Energy-Aware Adaptation for Mobile Applications." (1999)

Bibliography (part 2)

- Gu, Xiaohui, Klara Nahrstedt, Alan Messer, Ira Greenberg, and Dejan Milojevic. "Adaptive Offloading Inference for Delivering Applications in Pervasive Computing Environments." (2003)
- Gummadi, Ramakrishna, Omprakash Gnawali, and Ramesh Govindan. "Macro-Programming Wireless Sensor Networks Using Kairos." (2005)
- Hunt, Galen C, and Michael L Scott. "The Coign Automatic Distributed Partitioning System." (1999)
- Kothari, Nupur, Ramakrishna Gummadi, Todd Millstein, and Ramesh Govindan. "Reliable and Efficient Programming Abstractions for Wireless Sensor Networks." (2007)
- Kotsifakou, Maria, Prakalp Srivastava, Matthew D. Sinclair, Rakesh Komuravelli, Vikram S. Adve, and Sarita V. Adve. "HPVM: Heterogeneous Parallel Virtual Machine." (2018)
- Lakshmanan, G. T., Y. Li, and R. Strom. "Placement Strategies for Internet-Scale Data Stream Systems." (2008)
- Mach, P., and Z. Becvar. "Mobile Edge Computing: A Survey on Architecture and Computation Offloading." (2017)
- Mottola, Luca, and Gian Pietro Picco. "Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art." (2011)
- Neubauer, Matthias, and Peter Thiemann. "From Sequential Programs to Multi-Tier Applications by Program Transformation." (2005)
- Newton, R., G. Morrisett, and M. Welsh. "The Regiment Macroprogramming System." (2007)
- Newton, Ryan, Sivan Toledo, Lewis Girod, Hari Balakrishnan, and Samuel Madden. "Wishbone: Profile-Based Partitioning for SensorNet Applications." (2009)
- Ni, Yang, Ulrich Kremer, Adrian Stere, and Liviu Iftode. "Programming Ad-Hoc Networks of Mobile and Resource-Constrained Devices." (2005)
- Pietzuch, P., J. Ledlie, J. Shneidman, M. Roussopoulos, M. Welsh, and M. Seltzer. "Network-Aware Operator Placement for Stream-Processing Systems." (2006)
- Qiao, Y., R. Nolani, S. Gill, G. Fang, and B. Lee. "ThingNet: A Micro-Service Based IoT Macro-Programming Platform over Edges and Cloud." (2018)

Bibliography (part 3)

- Shuja, Junaid, Abdullah Gani, Kwangman Ko, Kyoungyoung So, Saad Mustafa, Sajjad A. Madani, and Muhammad Khurram Khan. "SIMDOM: A Framework for SIMD Instruction Translation and Offloading in Heterogeneous Mobile Architectures." (2018)
- Weinsberg, Yaron, Danny Dolev, Tal Anker, Muli Ben-Yehuda, and Pete Wyckoff. "Tapping into the Fountain of CPUs: On Operating System Support for Programmable Devices." (2008)
- Weisenburger, Pascal, Johannes Wirth, and Guido Salvaneschi. "A Survey of Multitier Programming." (2020)
- Welsh, Matt, and Geoff Mainland. "Programming Sensor Networks Using Abstract Regions." (2004)
- Whitehouse, Kamin, Cory Sharp, Eric Brewer, and David Culler. "Hood: A Neighborhood Abstraction for Sensor Networks." (2004)
- Wienke, Sandra, Christian Terboven, James C. Beyer, and Matthias S. Müller. "A Pattern-Based Comparison of OpenACC and OpenMP for Accelerator Computing." (2014)
- Xu, D., Y. Li, X. Chen, J. Li, P. Hui, S. Chen, and J. Crowcroft. "A Survey of Opportunistic Offloading." (2018)
- Zaharia, Matei, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, and Ion Stoica. "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling." (2010)
- Zaharia, Matei, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J Franklin, Scott Shenker, and Ion Stoica. "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing." (n.d.)
- Zdancewic, Steve, Lantian Zheng, Nathaniel Nystrom, and Andrew C. Myers. "Secure Program Partitioning." (2002)
- Zhou, Bowen, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, and Rajkumar Buyya. "An Online Algorithm for Task Offloading in Heterogeneous Mobile Clouds." (2018)
- Zhou, Bowen, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, Satish Narayana Srirama, and Rajkumar Buyya. "MCloud: A Context-Aware Offloading Framework for Heterogeneous Mobile Cloud." (2017)