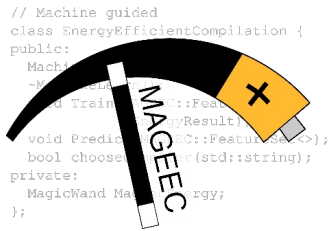


Machine Guided Energy Efficient Compilation

Jeremy Bennett, CEO Embecosm



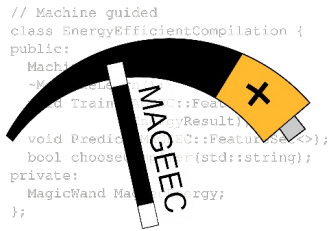
The Impact of Different Compiler Options on Energy Consumption



James Pallister
Embecosm / University of Bristol

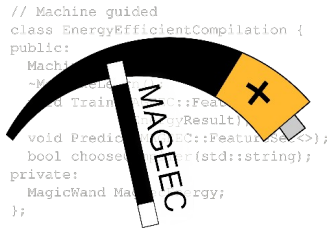
Simon Hollis
University of Bristol

Jeremy Bennett
Embecosm



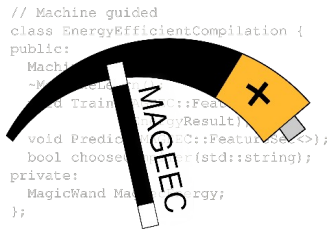
Motivation

- Compiler optimizations are claimed to have a large impact
 - performance
 - code size
 - energy usage
- No *extensive* study prior to this considering
 - different benchmarks
 - individual optimizations
 - different platforms
- This work looks at the effect of
 - many different optimizations
 - 10 benchmarks
 - 5 platforms.
- Over 200 Optimization passes covered by around 150 flags



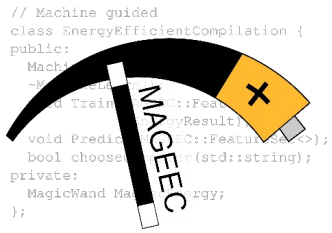
Key Components

- Importance of benchmarks
 - new set defined for embedded systems
- Choice of platforms
 - Epiphany, XMOS and 3 flavors of ARM
- How to explore 2^{150} combinations of options
 - fractional factorial design
- Energy measuring hardware
 - *not* simulation
- Result: Large dataset of extensive results



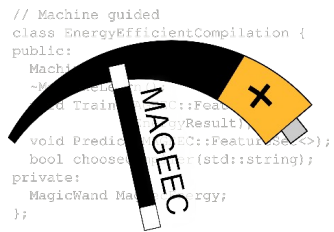
Results

- Time \approx Energy
 - true for simple pipelines
 - mostly true for complex pipelines
 - good first approximation
- Optimization is very unpredictable
 - difficult to model the interactions between optimizations
- There is only modest commonality
 - some common options for a single architecture
 - some common options within the ARM family
 - sometimes common options across a benchmark



Results

- Time \approx Energy
 - true for simple pipelines
 - mostly true for complex pipelines
 - good first approximation
- Optimization is very unpredictable
 - difficult to model the interactions between optimizations
- There is only modest commonality
 - some common options for a single architecture
 - some common options within the ARM family
 - sometimes common options across a benchmark
- **Summary: You can't predict which optimizations are best**



What is MAGEEC?



Today we optimize for
speed or space


```
// Machine guided
class EnergyEfficientCompilation {
public:
    MagicWand Wand;
    ~MagicWand() { Wand.Destroy(); }
    void Predict(C::FeatureSet& f, C::Result& r);
    bool choose(C::FeatureSet& f, C::Result& r, std::string s);
private:
    MagicWand Wand;
    MagicWand Wand;
};
```



What is MAGEEC?

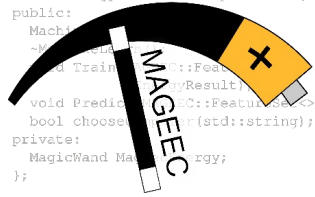


Today we optimize for
speed or space



What if we could optimize for
energy usage?

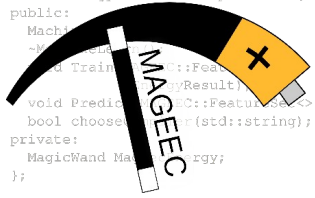

```
// Machine guided
class EnergyEfficientCompilation {
public:
    Machi
    ~Mach
    void Train(C::Feat
    void Predict(C::Feat
    bool choose(std::string);
private:
    MagicWand Ma
};
```



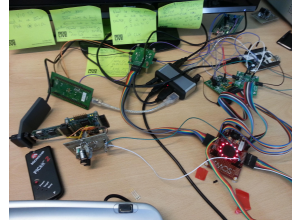
Research into
modeling energy usage

How We Got Here

```
// Machine guided
class EnergyEfficientCompilation {
public:
    MagicWand
    ~MagicWand() {}
    void Train(C::FeatureSet<>);
    void Predict(C::FeatureSet<>);
    bool choose(C::FeatureSet<>);
private:
    MagicWand MagicWand;
};
```



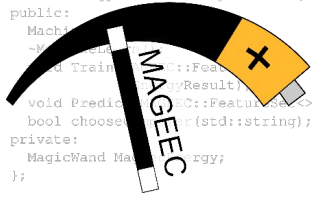
Research into
modeling energy usage



Energy
measurement

How We Got Here

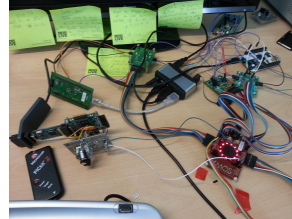
```
// Machine guided
class EnergyEfficientCompilation {
public:
    Machi
    ~Mach
    void Train(C::Feature, C::Result)
    void Predict(C::Feature, C::Result)
    bool choose(C::Result, C::Result);
private:
    MagicWand MagicWand;
};
```



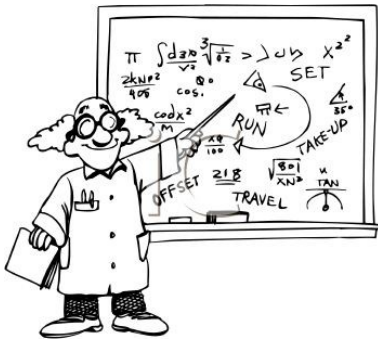
How We Got Here



Research into
modeling energy usage

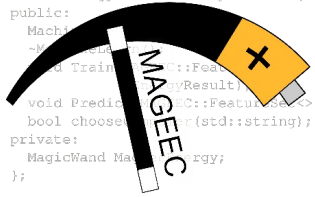


Energy
measurement



Research into feedback
directed optimization

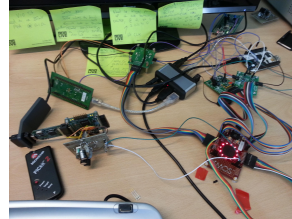
```
// Machine guided
class EnergyEfficientCompilation {
public:
    Machi
    ~Mach
    void Train(C::Feature
    void Predict(C::Feature
    bool choose
private:
    MagicWand Ma
};
```



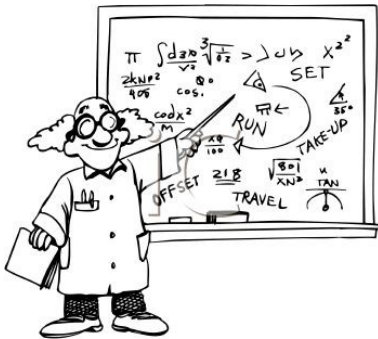
How We Got Here



Research into
modeling energy usage



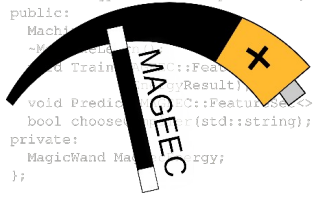
Energy
measurement



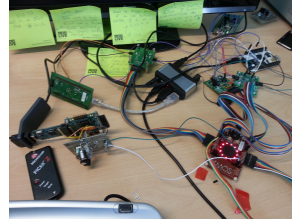
Research into feedback
directed optimization



```
// Machine guided
class EnergyEfficientCompilation {
public:
    MagicWand
    ~MagicWand() {}
    void Train(C::FeatureSet& fset, C::Result& r) {
        void Predict(C::FeatureSet& fset, C::Result& r) {
        bool choose(C::Result& r, std::string& s);
    private:
        MagicWand MagicWand;
    };
};
```

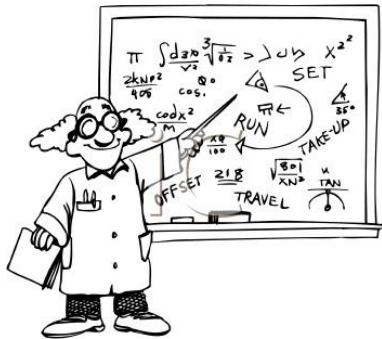


Research into
modeling energy usage



Energy
measurement

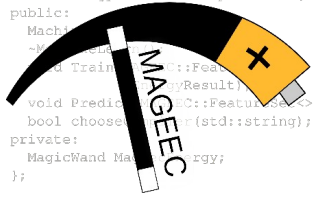
How We Got Here



Research into feedback
directed optimization



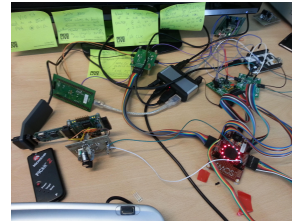
```
// Machine guided
class EnergyEfficientCompilation {
public:
    Machi
    ~Mach
    void Train(C::Fea
    void Predict(C::Featur
    bool chooseer(std::string);
private:
    MagicWand Ma
    Energy;
};
```



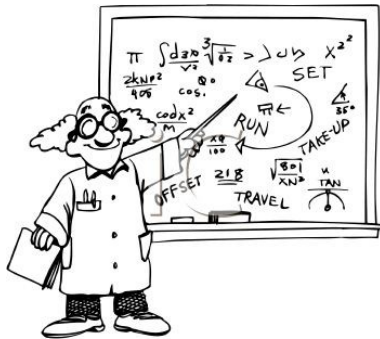
How We Got Here



Research into
modeling energy usage



Energy
measurement



Research into feedback
directed optimization



```
// Machine guided
class EnergyEfficientCompilation {
public:
    Machi
    ~Mach
    void Train(C::Fea
    void Predict(C::Featur
    bool chooseer(std::string);
private:
    MagicWand Ma
    Energy;
};
```

```
// Machine guided
class EnergyEfficientCompilation {
public:
    Machi
    ~Mach
    void Train(C::Feat
    void Predict(C::Feat
    bool choose(C::Feat
private:
    MagicWand Ma
};
```

What's New?



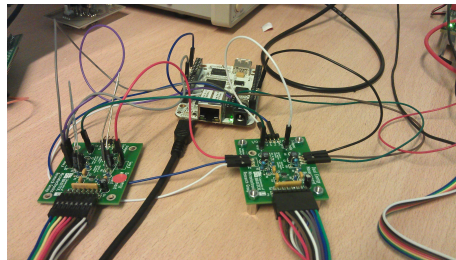
Objective is energy optimization


```
// Machine guided
class EnergyEfficientCompilation {
public:
    MagicWand Wand;
    ~MagicWand() { Wand.Destroy(); }
    void Predict(MagicWand::Result);
    bool choose(MagicWand::Result);
private:
    MagicWand Wand;
};
```

What's New?



Objective is energy optimization



Energy measured *not* modeled

```
// Machine guided
class EnergyEfficientCompilation {
public:
    MachineEfficientCompilation() {}
    ~MachineEfficientCompilation() {}
    void PredictMachineEfficiency(const FeatureSet& f);
    bool chooseMachine(const std::string& m);
private:
    MagicWand MachineEfficiency;
};
```

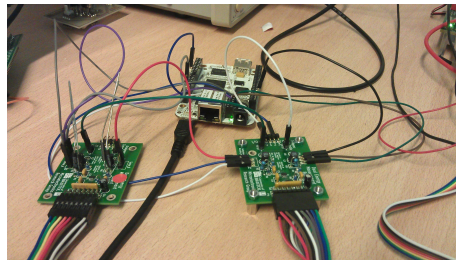
What's New?



Objective is energy optimization



Generic framework: GCC *and* LLVM initially



Energy measured *not* modeled

```
// Machine guided
class EnergyEfficientCompilation {
public:
    Machine
    ~Machine() {
        void Train(C::Feature, Result);
        void Predict(C::Feature, Result);
        bool choose(C::Feature, Result);
    private:
        MagicWand MagicWand;
    };
};
```

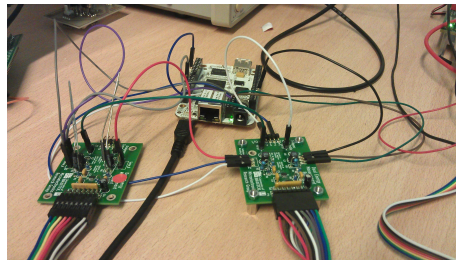
What's New?



Objective is energy optimization



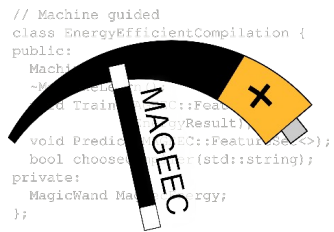
Generic framework: GCC *and* LLVM initially



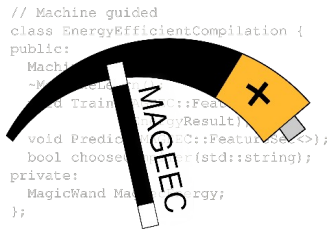
Energy measured *not* modeled



Working system, not research prototype

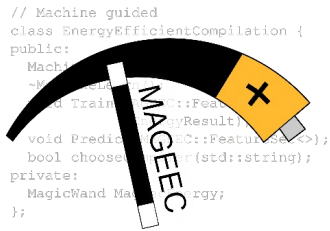


Implementation



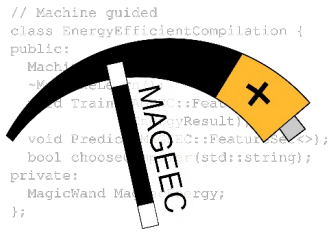
Our Plan

- Implement MILEPOST concepts in a generic way.



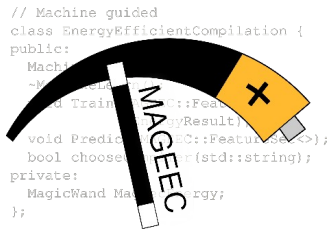
Our Plan

- Implement MILEPOST concepts in a generic way.
- Train and evaluate based on real hardware energy measurements and existing passes.

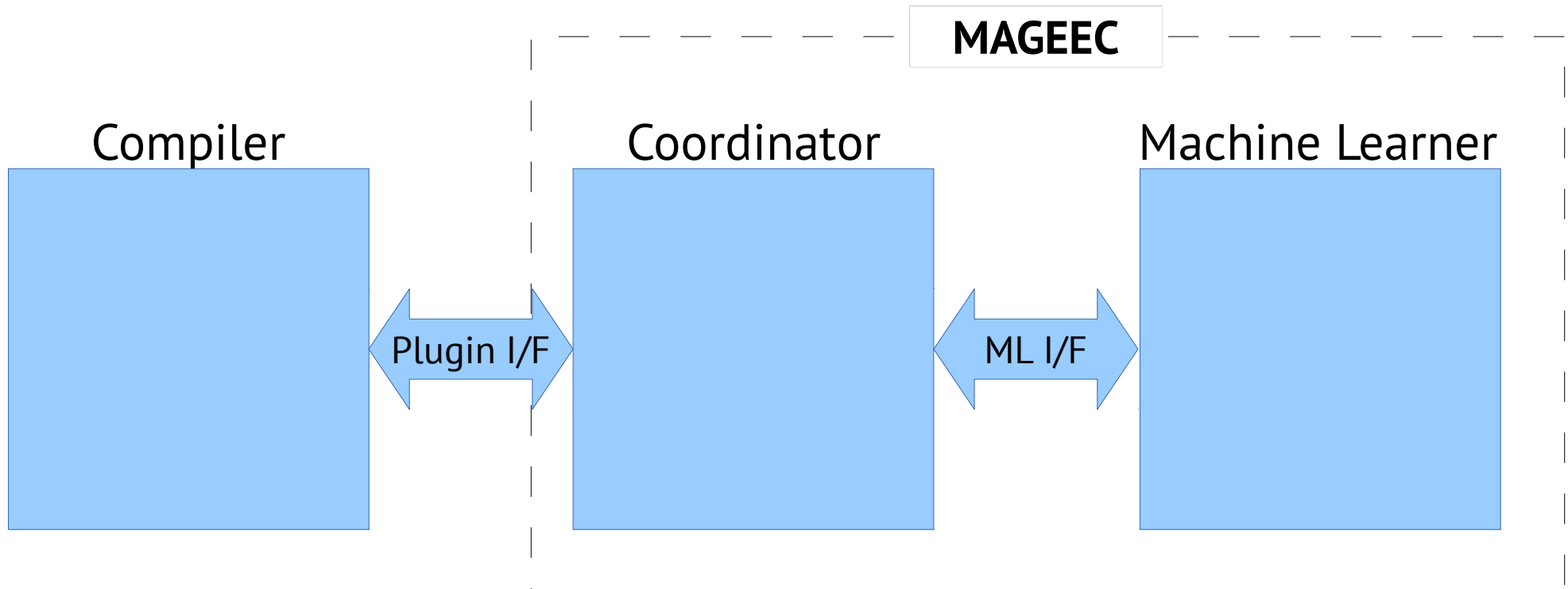


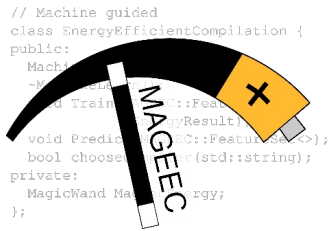
Our Plan

- Implement MILEPOST concepts in a generic way.
- Train and evaluate based on real hardware energy measurements and existing passes.
- Write and evaluate optimization passes specifically for energy efficiency (Jörn Rennecke).

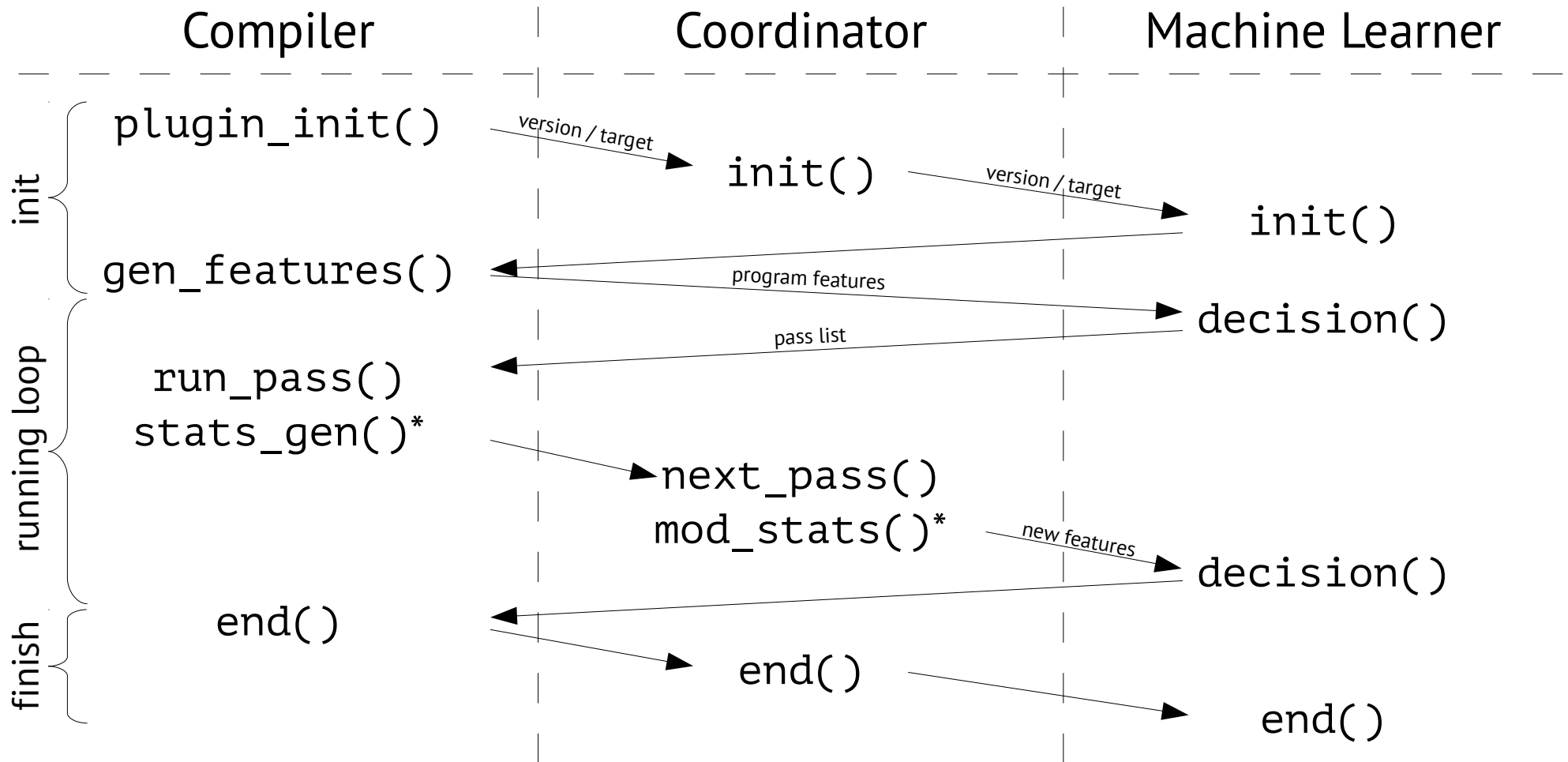


Overall Design

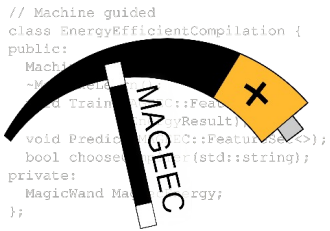




Overall Design



* may be `gen_features()`



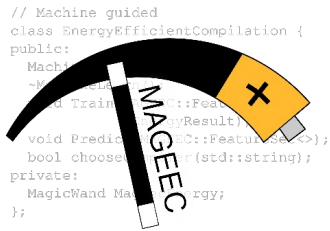
Community Involvement

Category:Planning - MAGEEC
 This category holds all documents related to the planning stage of the MAGEEC project. A PDF version of core documents is available.

Category:Design - MAGEEC
 This page contains all the design-related documents.

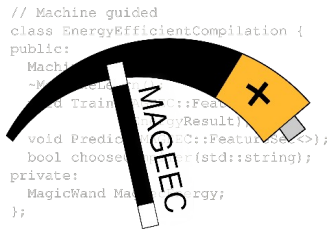
Interface Flow - MAGEEC
 This page describes the interfaces and basic flow of how the three components of MAGEEC will communicate. Between the components are the direction of communication along with what data is communicated.

Compiler	MAGEEC	ML	Comment
plugin_init	-> v/target		
	init	->	
		init	
		<-	
	needpass?		(only if needed)
genpasslist	->		
	build/load list	-> list + vtarget	
		load_db	(based of v/target)
		<-	
gen_features	-> src/name		(Need to define features for here + what to do if not avail)
		-> features	
		decision	(Initial decision, pass list of N passes for each pass)



Further Reading

- Energy measuring and modeling
 - *The software drained my battery.* Kerstin Eder & Jeremy Bennett, NMI Yearbook 2012, www.embecosc.com/resources/articles/#EAR12.
- MILEPOST GCC - Feedback directed optimization
 - ctuning.org/milepost-gcc
- Measurement of compiler energy usage
 - *Identifying Compiler Options to Minimize Energy Consumption for Embedded Platforms.* James Pallister, Simon Hollis, Jeremy Bennett arxiv.org/abs/1303.6485
- MAGEEC
 - mageec.org



Thank you

mageec.org

www.embecosm.com

cs.bris.ac.uk