Goals and Design Principles

- End-to-end experimental platform for MT (SAMT)
  - Starting with alignment models and phrase pairs
  - Generate SAMT rules, run translation and MER
  - Use HDFS + SVN as an experimentation platform
- Tradeoff memory for disk (HDFS) when using large models
- Adapt existing code via Hadoop streaming only
- Be good shared workqueue citizens

```
run_samt.pl
--params_dir
    ../examples/iwslt/
--expname=test1
--existing_phases=
    extract,filter
--mer --iter_limit 8
```

```
/iwslt/test1/extract_rules
/iwslt/test1/filter_rules_dev
/iwslt/test1/filter_rules_test
/iwslt/test1/phrases
/iwslt/test1/src_and_refs_dev
/iwslt/test1/src_and_refs_test
/iwslt/test1/translation_dev
/iwslt/test1/translation_test
/iwslt/test1/mer
/iwslt/test1/merged0
/iwslt/test1/merged1
/iwslt/test1/merged2
```

SAMT in MapReduce

RuleExtraction: Stream Input $e, f, phrases(e, f), \pi$

Map: Generate rules r: γ, α, lhs: key=ul(γ), value=r
Reduce: Calculate some features in ϕ based on rule block

RuleFiltering: Stream Input: rules, Side Input: dev. corpus

Map: Rules for sentence: key=*sno* value=rule, special counts for ϕ
Reduce: Finish generating features ϕ, add system rules

LMFiltering: Stream Input: N-gram LM, Side Input: filtered rules

Map: Output n-grams for rule: key=*sno* value=*ngram*
Reduce: Output SRI-LM for each *sno*

Translation: Stream Input: Corpus to translate (load balanced)

Map: Run Translation: key=*sno* value=nbest-list for *sno*

Merge (for MER): Stream Input: Translation output

Reduce: Removes duplicate translations for each *sno*

MER: Stream Input: Merged n-best lists: Map: Run MER