

# CloudI



## A Cloud as an Interface

Erlang User Conference, Stockholm, Sweden  
November 12, 2009

Michael Truog  
mjtruog@gmail.com

### What is Cloudi?

1. Private Cloud Computing Framework
2. Fault-tolerant Work Processing
3. Dynamic Load Balancing and Scheduling
4. Ordered Work Input/Output
5. Distributed Execution of C/C++ Work
6. The Future

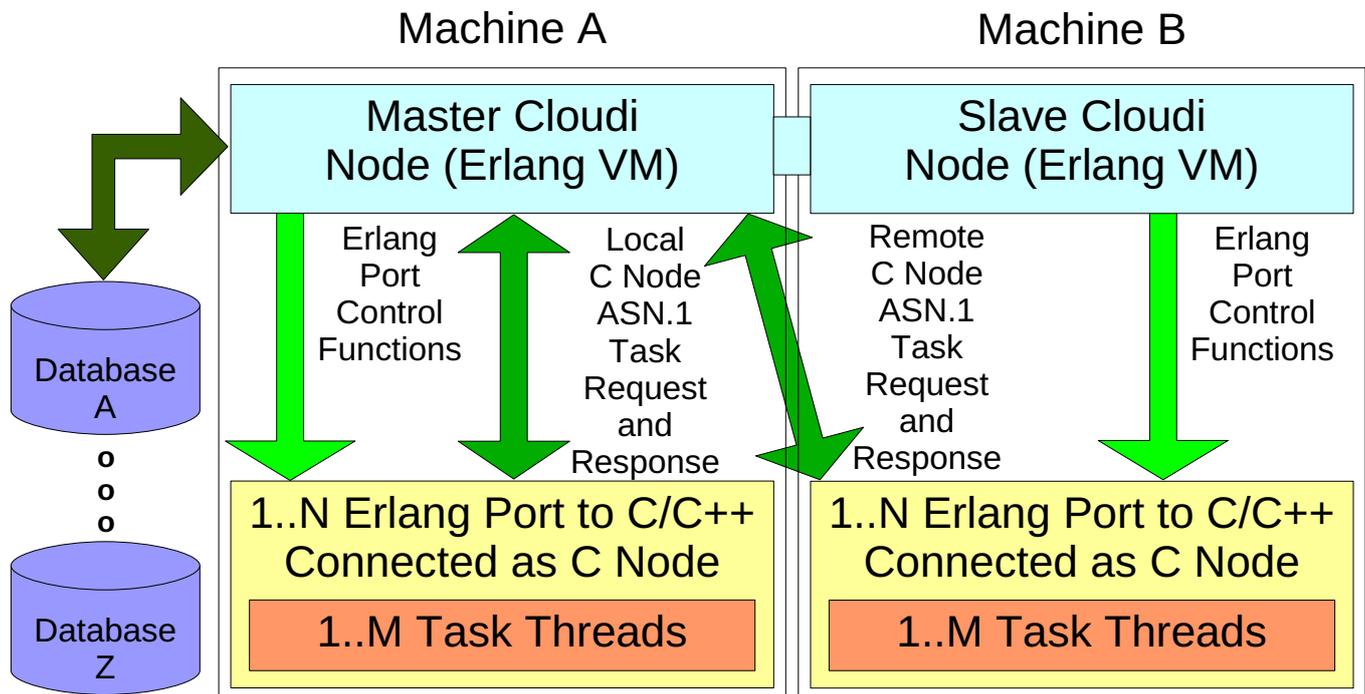
# A Private Cloud Computing Framework

- Provides an open-source cloud
  - BSD License
- An alternative to paying for a black-box commercial cloud
  - Internal processing is secure processing
- Creates a stable distributed processing environment from any available Linux machines

## Fault-tolerant Work Processing

- Erlang/OTP coordinates all work allocation, execution, and data output
- Any crash of C/C++ code is handled
  - Any signals, including uncatchable signals
- Uses Erlang Port processes subscribing to the cloud as Erlang C Nodes
  - Fault-tolerance overhead is currently 0.172 ms/task locally and 0.347 ms/task remotely (based on `cloud_job_latency` test results)

# Fault-tolerant Work Processing . . .



## Dynamic Load Balancing and Scheduling

- Workers are ideally stateless and form a pool of workers in the cloud
- Cloudi adjusts the task size based on the task execution time that is requested
  - Convergence is slow to avoid problems with unstable work processing
- Cloudi verifies that work is loaded
  - During work allocation
  - After node reconnection

## Ordered Work Input/Output

- The Erlang work module enforces an order on the work task input
- Cloudi maintains the task input order when collecting output so data is stored in the same order
- Work processing is paused when excessive data accumulation occurs

## Distributed Execution of C/C++ Work

- One “do\_work” function is required in a dynamic library for the C/C++ work
  - Loaded when Cloudi requests it
- Six Erlang functions within the work module provide work task specification
  - The functions define the task size (float value in range (0..1)) and task data (binary data)
- Any Erlang data module can handle output
  - PostgreSQL, MySQL, memcached, Tokyo Tyrant

## The Future

- Replicated Cloudi instances can be used for failover (needs management application)
  - Failover uses separate epmd processes for local name registration
- More databases will be supported
- More fault-tolerance testing
- Download Cloudi @ <http://cloudi.org/>
  - Version 0.0.8 alpha is now available!

Questions?