



AJUSTE FINO AUTOMATIZADO UTILIZANDO COMPUTAÇÃO EM NUVEM

Pedro Bruel
phrb@ime.usp.br

29 de Setembro de 2015

Departamento de Ciência da Computação do IME, USP
MAC5910 - Programação para Redes de Computadores

RETOMANDO

MOTIVAÇÕES

Arquiteturas multi-core e co-processadores:

- Grande variedade e disponibilidade
- **Potencial** para alto desempenho
- Cada vez mais **difícil realizar** esse potencial

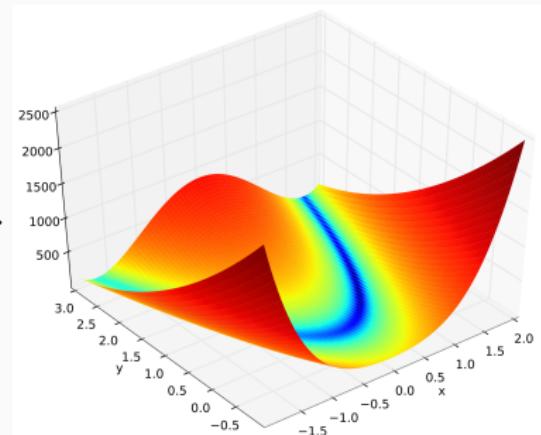


¹Imagen: wccftech.com/amd-cuddling-video-cards-charity-twitter/

UMA SOLUÇÃO: AUTOTUNING

Configurações e Otimizações

Espaço de Busca



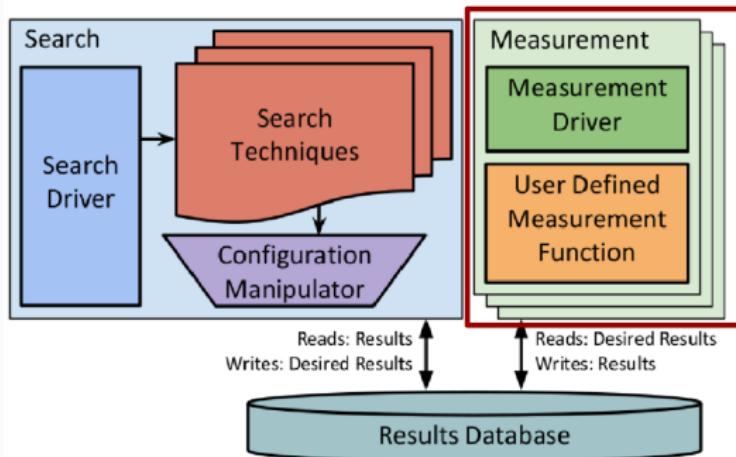
Problema: O processo pode ser muito demorado!

AUTOTUNING: ALGUNS SISTEMAS

Sistema	Domínio	Método de Busca
ATLAS	Álgebra Linear (Matrizes Densas)	Exaustivo
Code Perforation	Compilador	Exaustivo e Simulated Annealing
FFTW	Transformada Rápida de Fourier	Exaustivo e Programação Dinâmica
OSKI	Álgebra Linear (Matrizes Esparsas)	Exaustivo e Heurísticas
Active Harmony	Ambiente de Execução	Nelder-Mead
PATUS	Computação Estêncil	Nelder-Mead e Evolucionário
SPIRAL	Algoritmos para DSP	Pareto Active Learning
Dynamic Knobs	Ambiente de Execução	Control Theory
MilepostGCC / cTuning	Compilador	Variáveis Aleatórias e Banco de Dados
SEEC / Heartbeats	Ambiente de Execução	Control Theory
INSIEME	Compilador	Differential Evolution
StreamJIT	Compilador Comensal (Dataflow)	OpenTuner
Petabricks	Linguagem de Programação	Bottom-Up Evolutionary
OpenTuner(*)	Autotuning Framework	Vários

Tabela 1: Sistemas para Autotuning.

AUTOTUNING: OPENTUNER



Medições são sempre feitas **localmente** e **sequencialmente**

²Imagem: Ansel, Jason, et al. "Opentuner: An extensible framework for program autotuning." Proceedings of the 23rd ICPAC. ACM, 2014.

PROPOSTA PARA O ARTIGO

OBJETIVOS

Modificar o arcabouço OpenTuner para que seja possível realizar medições distribuídas na nuvem

Modificar o arcabouço OpenTuner para que seja possível realizar medições distribuídas na nuvem

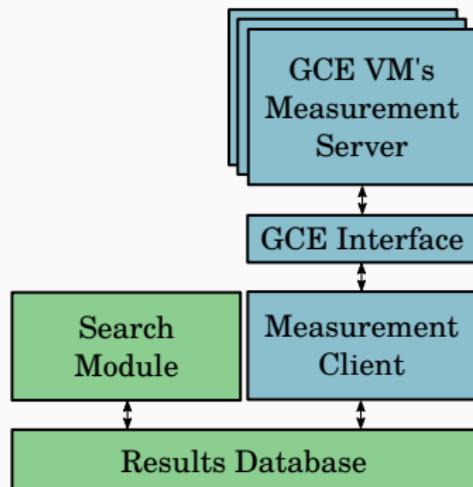
RQ1: Como normalizar medições de desempenho feitas na nuvem?

Modificar o arcabouço OpenTuner para que seja possível realizar medições distribuídas na nuvem

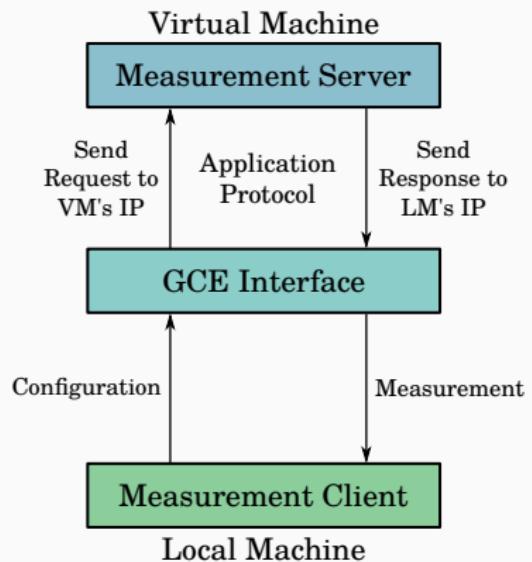
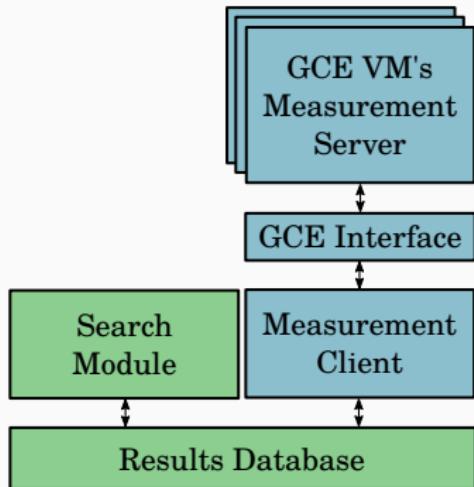
RQ1: Como normalizar medições de desempenho feitas na nuvem?

RQ2: Para que tipo de problema é vantajoso utilizar os recursos da nuvem?

CLIENTE E SERVIDOR



CLIENTE E SERVIDOR



SERVIDOR DE ECO: PYTHON

```
#!/usr/bin/env python

import socket

TCP_IP = ''
TCP_PORT = 8080
BUFFER_SIZE = 1024

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((TCP_IP, TCP_PORT))
s.listen(1)

conn, addr = s.accept()

while 1:
    data = conn.recv(BUFFER_SIZE)
    if not data: break
    conn.send(data)
```

```
class MeasurementClient(MeasurementDriver):
    # Reads DesiredResults and requests VMs
    # in the cloud to compute Results.
    def __init__(self,
                 measurement_interface,
                 input_manager,
                 **kwargs):
        super(MeasurementDriver, self).__init__(**kwargs)
        # Instantiate and configure the VM Measurement
        # Servers using the GCE interface.

    def process_all(self):
        # Process all results in the
        # database, by sending requests to
        # VMs in the cloud and waiting
        # for responses.
```

NORMALIZAÇÃO DE RESULTADOS

RQ1: Como aplicar os resultados obtidos em máquinas virtuais diferentes da máquina local?

- Autotuning do **modelo de desempenho**
- **Combinar** resultados diferentes
- **Simular** a máquina local
- Executar o autotuner na **nuvem**

EXPERIMENTOS E VALIDAÇÃO

RQ2: Para quais tipos de problema é vantajoso utilizar os recursos da nuvem?

O conjunto de experimentos deve expor características de desempenho na nuvem, e permitir comparar as duas versões do OpenTuner

Retomando:

- MeasurementClient
- MeasurementServer
- Interface com a Google Compute Engine
- Protocolo de Aplicação
- Normalização de Resultados
- Avaliação de desempenho

CRONOGRAMA

AP2 & PR 25/09	AR1 16/10	AP3 & AR2 13/11	AR3 25/11
Paper proposal. 10-minute presentation. Preliminary experiments at GCE. High and low-level views of the implementation.	First version of the paper. Measurement Client implementation. GCE Interface implementation. Measurement Server implementation.	Second version of the paper. 20-minute presentation. Result normalization techniques. Benchmark choice. Application protocol full specification.	Final version of the paper. Experiments, results and analysis.

Tabela 2: Cronograma do Projeto.

OBRIGADO! PERGUNTAS?



AJUSTE FINO AUTOMATIZADO UTILIZANDO COMPUTAÇÃO EM NUVEM

Pedro Bruel
phrb@ime.usp.br

29 de Setembro de 2015

Departamento de Ciência da Computação do IME, USP
MAC5910 - Programação para Redes de Computadores