# Communication Theory in the Cloud: The Transformative Power of Cheap Utility Computing

#### Matthew C. Valenti

West Virginia University

Jan. 30, 2012

This work supported by the National Science Foundation under award CNS-0750821.

Valenti

# About Me

- From the state of Maryland, in the United States.
- Educated at
  - Virginia Tech
  - Johns Hopkins University.
- Worked as an Electronics Engineer at the U.S. Naval Research Laboratory.
- Professor at West Virginia University.





- 1 The Case for Cloud Computing
- 2 How Our CC Infrastructure Works
- 3 Applications of CC
- A Case Study
- **5** Conclusions

#### Outline

#### The Case for Cloud Computing

2 How Our CC Infrastructure Works

#### 3 Applications of CC

#### A Case Study

#### 5 Conclusions

# The Last Time I Was in New Zealand...

- Nov. 1998.
- 266 MHz Pentium II was state of the art.
- This figure took one month to generate!
- M.C. Valenti, "Turbo codes and iterative processing," in *Proc. IEEE New Zealand Wireless Commun. Symp.*, (Auckland, New Zealand), Nov. 1998.



# Since Then...

- Thanks to Moore's law, the cost per FLOP has decreased.
  - $\bullet\,$  However, single processor speeds have only gone up  $\sim 10\times.$
  - The push is towards more **cores**, rather than faster **clocks**.
  - **Clusters** of multi-core computers have become inexpensive, as has **utility** computing.
- Communication theory research could benefit from all this cheap, massively-parallel processing.
  - Simulation.
  - Optimization.
  - Anything that is embarrassingly parallelizable.
- However, communication theorists should not have to spend their time writing code for multi-core architectures.
  - Instead, computing should be provided to researchers as an easy-to-access **cloud computing** service.

# Utility and Cloud Computing

Utility computing providers are computing landlords.

- The providers own and maintain the infrastructure.
- They rent out processing capacity to their customers.
- Metered computing: analogous to electric power.
- Resources often virtualized and shared by multiple tenants.
- Example: Amazon Elastic Compute Cloud (\$2.04 USD/day).

**Cloud computing** not only provides raw computing power, but also hosts the applications that use the power.

- Specific applications supported rather than just raw computing power.
- Applications accessed via a web browser (usually).
- User data typically stored on provider's filesystems.
- Underlying computing infrastructure concealed from user.
- Key example: gmail.

# A Computing Cloud for Communication Theory

- Under NSF funding, we have created a cloud-computing resource to support the specific needs of the communication theory research community.
- The goals of our project are to:
  - Be compatible with Matlab.
  - Provide 100 1000 times speedup relative to a single PC.
  - Be accessible by the research community through a web-interface.
  - Support multiple open-source *projects*, with code contributed from the CT community <sup>1</sup>.
  - Allow the CT community to *donate* idle CPU cycles through the use of *volunteer computing*.
- Our philosophy is borrowed from Seymour Cray, "My guiding principle has always been simplicity. Only include features which are absolutely necessary."

<sup>&</sup>lt;sup>1</sup>Google code page: http://code.google.com/p/iscml/

#### Outline

#### The Case for Cloud Computing

#### 2 How Our CC Infrastructure Works

#### 3 Applications of CC

#### A Case Study

#### 5 Conclusions

# The User Interface

mcvalenti | Settings | Sign out

Jobs	Data files	:											
Delete Suspend Resume Add JobProjects													
All. None													
14 of 4 1.4 of 4 1.6 99													
	JobId	Project	Name	Upload time	Status								
	61	LDPC	DVBS2_3BY5_LONG	Jan 20	Queued								
	60	Outage	Outage_2_44_S1_SS0_1	Jan 20	Suspended								
	59	Outage	Outage_4_11_S0_SS1_2	Jan 20	20% Done								
	57	LDPC	DVBS2_1BY4_SHORT	Jan 20	Done								
🛞 🛞 1-4 of 4 🛞 🛞													

You have used 1.20 units of your 100 units of runtime (1.20%).

©2012 WCRL - Terms - Privacy Policy

The web interface is developed using the **Google Web Toolkit** (GWT), which is the same technology used to implement *gmail*.

http://wcrlcluster.csee.wvu.edu:8180/ldpc/

# Some Terminology

- A **project** is a collection of open-source code for solving a specific CT problem, e.g. simulate a LDPC code, optimize a wireless network.
- A **job** is a project-specific request to generate results, e.g. generate a BER curve, determine optimal network parameters. The request is embodied by a **job file**.
- A **task** is a a work-unit associated with a job that can be run on a single processor in a small increment of time (e.g., 5 minutes). The task is embodied by a **task file**.
- A **worker** is a process running on a computing cluster or grid that services tasks.
- A **queue** is an entity that holds job or task files that are either waiting to be serviced (input queue), being serviced (running queue), or completed (output queue).



The user has a job queue on the server for each project



The user has a job queue on the server for each project.

Each job queue has three components: input, running, and output.





The project-specific **job manager** places several task files into the user's *task input* queue

server





The project-specific job manager places several task files into the user's task input queue, and moves the job file into the project's job running queue.

server





The system can support multiple users. Each user has a set of job queues for each project it is subscribed to, and a set of task queues.



server





A **task manager** copies task files to the global *task input* queue.





A task manager copies task files to the global task input queue.

The task files are moved to the user's running task queue.























Each worker will randomly pick a task file from the global task input queue.

The workers move the task files to the global *running* queue.

Each worker runs its task to completion. This typically takes 5 minutes.





























#### cluster



The job manager checks to see if the stopping criteria has been met.

If it has been met, the task file is moved the the output task queue.

If it has not been met, then the job manager creates more task files, and the process repeats...



# Volunteer Computing

**Volunteer computing** initiatives create a computing grid using the idle cycles of volunteered computers.

• Examples: Folding@home, SETI@home (Driven by *BOINC* software).

Issues:

- Should only run when computer is not in use. Solution: Screensaver (windows, mac) or low-priority process (linux).
- $\ \ 2. \ \ Should \ \ work \ \ on \ \ any \ \ common \ OS.$

Solution: Virtualization.

3. Volunteered computer should not require a copy of Matlab. Solution: The Matlab Compiler.

Volunteer computing is facilitated by using **Frontier**, a software solution by Parabon, Inc.

How Our CC Infrastructure Works

# How to Involve Volunteer Computing



#### Outline

- The Case for Cloud Computing
- 2 How Our CC Infrastructure Works

# 3 Applications of CC

A Case Study

#### 5 Conclusions

# Applications to Research

- Monte Carlo Simulation
  - Can execute simulations written using the *Coded Modulation Library* (CML)<sup>2</sup> in parallel.
  - Support for a wide range of modulations and error-control codes.
- Optimization of (Space-Time) Modulation
  - Genetic algorithms used to design a space-time code<sup>3</sup> and to optimize the symbol labeling of QAM<sup>4</sup>.
- Network Optimization
  - Optimize parameters related to the modulation, coding, and multiple-access method.
  - Applicable to both ad hoc and infrastructure (cellular) networks.
  - Infrastructure is well suited for exhaustive (brute-force) optimization methods.

 $^{2} http://code.google.com/p/iscml/$ 

<sup>3</sup>D. Torrieri and M.C. Valenti, "Efficiently decoded full-rate space-time block codes," *IEEE Trans. Comm.*, Feb. 2010.

<sup>4</sup>M.C. Valenti, R. Doppalapudi, and D. Torrieri, "A genetic algorithm for designing constellations with low error floors," in *Proc. CISS*, Mar. 2008.

Valenti

Communication Theory in the Cloud

Jan. 30, 2012 16 / 42

# **Cluster Performance**



# Applications to Teaching

- Wireless Networking (CPE 462)
  - Student groups form virtual wireless operators.
  - Given a budget to bid on spectrum and purchase infrastructure.
  - Design a cellular network by determining how many base stations to purchase, where to place them, and how to manage frequency resources.
  - Designs uploaded to the cluster, which analyzes performance.
- Communication Theory (EE 561)
  - Students optimize a signal set for operation over an AWGN channel.
  - Designs uploaded to the cluster, which analyzes and simulates the modulation.
- Coding Theory (EE 567)
  - Students design and optimize an LDPC code.
  - Designs uploaded to the cluster, which simulates the code in AWGN.

#### Outline

- The Case for Cloud Computing
- 2 How Our CC Infrastructure Works
- 3 Applications of CC
- A Case Study
  - 5 Conclusions

# Ad Hoc Networks



- Transmitters are randomly placed in 2-D space.
  - $X_i$  denotes 2-D location of  $i^{th}$  node.
  - Spatial model important (usually Poisson Point Process).
- Each node transmits to a random receiver.
  - Reference receiver located at the origin.
  - $|X_i|$  is distance to  $i^{th}$  node.
  - $X_0$  is location of reference transmitter.
  - M interfering transmitters,  $\{X_1, ..., X_M\}$ .

# **Frequency Hopping**



- Transmitters randomly pick from among L frequencies.
- A reference receiver will pick same frequency as the reference transmitter with probability p = 1/L.
- Interference avoidance.
- Preferred for ad hoc networks.

#### **SINR**

The performance at the reference receiver is characterized by the signal-to-interference and noise ratio (SINR), given by:

$$\gamma = \frac{g_0 \Omega_0}{\Gamma^{-1} + \sum_{i=1}^M I_i g_i \Omega_i}$$
(1)

where:

- $\Gamma$  is the SNR at unit distance.
- $g_i$  is the power gain due to fading (i.e. Rayleigh or Nakagami fading).
- $I_i$  is a Bernoulli indicator with  $P[I_i] = p$ .
- $\Omega_i = \frac{P_i}{P_0} 10^{\xi_i/10} ||X_i||^{-\alpha}$  is the normalized received power.
  - $P_i$  is the power of transmitter  $X_i$ .
  - $\xi_i$  is the dB shadowing gain (i.e. log-normal shadowing).
  - $\alpha$  is the path loss.

#### A Case Study

# **Outage Probability**

- An outage occurs when  $\gamma \leq \beta$ , where  $\beta$  is an SINR threshold.
- From (1), the outage probability is

$$\epsilon = P\left[\underbrace{\frac{g_0\Omega_0}{\Gamma^{-1} + \sum_{i=1}^{M} I_i g_i \Omega_i}}_{\gamma} \leq \beta\right]$$
$$= P\left[\underbrace{\beta^{-1}g_0\Omega_0 - \sum_{i=1}^{M} I_i g_i\Omega_i}_{\mathbf{Z}} \leq \Gamma^{-1}\right]$$
$$= P\left[\mathbf{Z} \leq \Gamma^{-1}\right] = F_{\mathbf{Z}}(\Gamma^{-1}).$$

• To find the outage probability, we should find an expression for the cdf of Z.

# **Rayleigh Fading**

• When all links are subject to Rayleigh fading,

$$F_{\mathsf{Z}}(z) = 1 - e^{-\beta z} \prod_{i=1}^{M} \frac{1 + \beta(1-p)\Omega_i}{1 + \beta\Omega_i}.$$

where it is assumed that:

- The reference transmitter is at unit distance,  $|X_0| = 1$ .
- There is no shadowing.

(2)

# Nakagami Fading

If the channel from the  $i^{th}$  node to the receiver is Nakagami-m with parameter  $m_i$ , then for integer  $m_0$ ,

$$F_{\mathsf{Z}}(z) = 1 - \exp\left\{\beta z \frac{m_0}{\Omega_0}\right\} \sum_{s=0}^{m_0-1} \left(\beta z \frac{m_0}{\Omega_0}\right)^s \sum_{r=0}^s \frac{z^{-r} V_r(\Psi)}{(s-r)!}$$

$$V_r(\Psi) = \sum_{\substack{\ell_i \ge 0\\ \sum_{i=0}^M \ell_i = r}} \prod_{i=1}^M U_{\ell_i}(\Psi_i)$$

$$U_\ell(\Psi_i) = \begin{cases} 1 - p(1 - \Psi_i^{m_i}), & \text{for } \ell = 0\\ \frac{p\Gamma(\ell + m_i)}{\ell!\Gamma(m_i)} \left(\frac{\Omega_i}{m_i}\right)^\ell \Psi_i^{m_i + \ell} & \text{for } \ell > 0 \end{cases}$$

$$\Psi_i = \left(\beta \left(\frac{m_0}{\Omega_0}\right) \left(\frac{\Omega_i}{m_i}\right) + 1\right)^{-1}, \quad \text{for } i = \{1, ..., M\}.$$

# An Example

- Reference (source) transmitter placed at distance  $|X_0| = 1$ .
- M = 50 interferers randomly placed in a circle of radius  $r_{max} = 4$ .
- L = 200 hopping frequencies, i.e. p = 1/200 = 0.005.
- $\beta = 3.7$  dB SINR threshold.
- Three fading models considered:
  - Rayleigh fading:  $m_i = 1$  for all *i*.
  - Nakagami fading:  $m_i = 4$  for all *i*.
  - Mixed fading:  $m_0 = 4$  for source and  $m_i = 1$  for interferers.
- Path-loss coefficient  $\alpha = 3$ .
- No shadowing.

# Example #1



Figure: Outage probability  $\epsilon$  as a function of SNR  $\Gamma$ . Analytical curves are solid, while • represents simulated values. The network geometry is shown in the inset, with the reference receiver represented by  $\star$  and interferers by •.

# Spatial Averaging

- $\epsilon$  is the outage probability conditioned on a specific network topology.
- We may want to average the outage probability with respect to the spatial distribution.
- The averaging can be done in closed-form for some, but not all, cases.
- When a closed-form solution does not exist, a Monte Carlo approach can be taken:
  - Draw N networks, each of size M.
  - Let  $\Omega_j$  be the set of  $\Omega_i$ 's for the  $j^{th}$  network.
  - Let  $F_{\mathsf{Z}}(z|\mathbf{\Omega}_j)$  be the cdf of Z for the  $j^{th}$  network.
  - $\bullet\,$  Take the average of the N cdfs

$$F_{\mathsf{Z}}(z) = \frac{1}{N} \sum_{j=1}^{N} F_{\mathsf{Z}}(z | \mathbf{\Omega}_j).$$

- As before, the outage probability is  $\epsilon = F_{\mathsf{Z}}(\Gamma^{-1}).$
- Shadowing can be modeled by including the factor 10<sup>ξi/10</sup> in each Ω<sub>i</sub>.
   For log-normal shadowing ξ<sub>i</sub> is zero mean Gaussian with variance σ<sub>s</sub><sup>2</sup>.

# Spatially-Averaged Outage



Figure: Average outage probability as function of the number of interferers M at SNR  $\Gamma = 10$  dB with SINR threshold  $\beta = 3.7$  dB in a mixed-fading environment. Curves are shown both with ( $\sigma_s^2 = 8$ ) and without shadowing. Results were obtained by averaging over  $N = 10\,000$  randomly generated networks.

# SINR Threshold

- Until now, we have picked the SINR threshold  $\beta$  arbitrarily.
- $\beta$  depends on the choice of modulation.
  - For *ideal* signaling

$$C(\gamma) = \log_2(1+\gamma)$$

 $\beta$  is the value of  $\gamma$  for which  $C(\gamma) = R$  (the code rate),

$$\beta = 2^R - 1$$

• For other modulations, the *modulation-constrained* capacity must be used.

# Modulation for Frequency Hopping

$$s_d(t) = \frac{1}{\sqrt{T_s}} e^{j2\pi dt/T_s}, \ d = 0, 1, \cdots, q-1$$

#### Orthogonal FSK

- Suitable for noncoherent reception.
- Reasonable energy efficiency.
- $\bullet\,$  Poor bandwidth efficiency because adjacent tones are  $1/T_s$  apart.

#### Nonorthogonal CPFSK

- Reduce bandwidth by using modulation index h < 1.
- Adjacent frequency tones are  $h/T_s$  apart.
- Continuous-phase constraint controls the spectrum.
- Transmitted  $x(t) = e^{j\phi}s_d(t)$  where phase  $\phi$  is accumulated

$$\phi = \phi' + 2\pi dh$$

# Modulation for Frequency Hopping

$$s_d(t) = \frac{1}{\sqrt{T_s}} e^{j2\pi dht/T_s}, \ d = 0, 1, \cdots, q-1$$

- Orthogonal FSK
  - Suitable for noncoherent reception.
  - Reasonable energy efficiency.
  - $\bullet\,$  Poor bandwidth efficiency because adjacent tones are  $1/T_s$  apart.
- Nonorthogonal CPFSK
  - Reduce bandwidth by using modulation index h < 1.
  - Adjacent frequency tones are  $h/T_s$  apart.
  - Continuous-phase constraint controls the spectrum.
  - Transmitted  $x(t) = e^{j\phi}s_d(t)$  where phase  $\phi$  is accumulated

$$\phi = \phi' + 2\pi dh$$

A Case Study

# Capacity of Noncoherent Binary CPFSK



(a) channel capacity versus  $\mathcal{E}_S/N_0$ 

(b) minimum  $\mathcal{E}_b/N_0$  versus coding rate

Reference: S. Cheng, R. Iyer Sehshadri, M.C. Valenti, and D. Torrieri, "The capacity of noncoherent continuous-phase frequency shift keying," in *Proc. Conf. on Info. Sci. and Sys. (CISS)*, (Baltimore, MD), Mar. 2007.

Valenti

Communication Theory in the Cloud

A Case Study

# Bandwidth of CPFSK



Valenti

Communication Theory in the Cloud

Jan. 30, 2012 33 / 42

# The Network Optimization Problem

• The objective is to maximize the *transmission capacity*, which for FH is

$$\tau = \frac{\eta R}{L} (1 - \epsilon) \lambda$$

where

- $\eta$  the (uncoded) modulation's spectral efficiency (bps/Hz).
- R is the rate of the channel code.
- L is the number of hopping frequencies.
- $\lambda = M/A$  is the *density* of the network, where A is the *area* of the network.
- $\bullet \ \epsilon$  is the spatially-averaged outage probability.
- Transport capacity is the area spectral efficiency.
  - Units of bps/Hz/ $m^2$ .
  - The rate that bits are successfully transmitted over 1 Hz BW and 1 square meter of area.
- Goal of the optimization is to pick R, h, and L that maximize  $\tau$ .

# Optimization Algorithm

Brute force optimization, assuming constant transmit power  $P_i = P_0, \forall i$ ,

- Create a set of  $\{\Omega_j\}$  corresponding to M networks, where each  $\Omega_j$  is found by placing N nodes  $\{X_i\}$  according to the desired spatial distribution and then computing  $\Omega_i = 10^{\xi_i/10} ||X_i||^{-\alpha}$  for each node, where  $\xi_i$  is drawn from the desired shadowing distribution.
- **②** For each L and  $\beta$  in a discretized set, perform the following **task**:
  - Compute the outage probability  $\epsilon$  averaged over the  $\{\Omega_j\}$ .
  - **②** For each h in a discretized set, determine the rate  $R = C(\beta)$ , which is the achievable rate of noncoherent CPFSK with modulation index h and SNR  $\beta$ .
  - For the set of (h, R) found in the last step, determine the normalized transmission capacity  $\tau = \eta R \lambda (1 \epsilon) / L$ .
- Solution Identify the (h, R, L) that maximizes  $\tau$ .
- **9** If desired, adjust the spacing and range of (h, R, L) and return to 2.

# **Example Optimization**

- Nakagami parameters:  $m_0 = 4$  and  $m_i = 1$  (mixed fading).
- Shadowing variance:  $\sigma_s^2 = 8$ .
- Path-loss exponent:  $\alpha = 3$ .
- Number of interferers: M = 50.
- Network dimensions:  $r_{min} = 0.25$  and  $r_{max} = 2$ .
- Number of networks generated: N = 100.

#### A Case Study

# Influence of h



Figure: Maximum transmission capacity  $\tau(h)$  (in  $bps/kHz - m^2$ ) as a function of the modulation-index h. For each value of h, the code rate R and number of frequency channels L are varied to maximize the TC.

#### A Case Study

# Influence of R



Figure: Maximum transmission capacity  $\tau(R)$  (in  $bps/kHz - m^2$ ) as a function of the code rate R. For each value of R, the modulation-index h and number of frequency channels L are varied to maximize the TC.

# Influence of L



Figure: Maximum transmission capacity  $\tau(L)$  (in  $bps/kHz - m^2$ ) as a function of the number of frequency channels L. For each value of L, the modulation-index h and code rate R are varied to maximize the TC.

#### **Optimization Results**

$r_{max}$	$\sigma_s^2$	$m_0$	$m_i$	L	R	h	$ au_{opt}$	$ au_{sub}$
2	0	1	1	31	0.61	0.59	15.92	3.31
		4	4	42	0.66	0.59	17.09	4.05
		4	1	36	0.65	0.59	19.82	4.13
	8	1	1	31	0.63	0.59	15.98	3.31
		4	4	41	0.66	0.59	17.43	4.04
		4	1	36	0.66	0.59	20.11	4.12
4	0	1	1	12	0.54	0.59	9.73	0.89
		4	4	15	0.50	0.59	10.65	1.12
		4	1	14	0.51	0.59	11.85	1.12
	8	1	1	12	0.53	0.59	9.41	0.89
		4	4	16	0.51	0.59	10.26	1.12
		4	1	14	0.52	0.59	11.46	1.12

Table: Results of the Optimization for M=50 interferers. The transmission capacity  $\tau$  is in units of bps/kHz- $m^2$ .  $\tau_{opt}$  is TC with the optimizer parameters, while  $\tau_{sub}$  is TC with (L,R,h)=(200,1/2,1).

# Conclusions

- Computing is a commodity.
- By aggregating donated CPU cycles and providing a web-interface, we have developed a computing resource for the communication theory community.
- The computing infrastructure is appropriate for:
  - Parallel Monte Carlo simulation.
  - Exhaustive optimization
- We will soon open the resource to the research community.
  - You may apply for an account<sup>5</sup>.
- Think about what **you** would do with 100-1000 times more computing power!

<sup>&</sup>lt;sup>5</sup>http://wcrlcluster.csee.wvu.edu:8180/ldpc/

# Thank you