

Ambient Cloud – Um Futuro Super Escalável

Beatriz Yoko Matsushita

Universidade Federal de São Carlos – Campus Sorocaba

yokobia@gmail.com

Resumo— O maior problema, hoje em dia, é a alocação de memória. Os recursos para armazenamento de dados são finitos, enquanto que o crescimento desses dados é infinito. Nesse contexto, entra o Ambient Cloud, que traz consigo o conceito de utilizar recursos ainda não explorados, muito mais baratos e escaláveis. São todos os dispositivos capazes de efetuar acesso à internet e possuem parcela de seus recursos ociosos.

Palavras-chave— ambient cloud computing, datacenters

I. INTRODUÇÃO

A Internet é um ambiente de muitas informações, sejam elas verdadeiras ou falsas, que aumentam cada vez mais, resultando em uma massa muito densa. Dia após dia, cresce o número de usuários que acessam a internet, requisitando ou armazenando informações. O fluxo de dados é enorme, demandando absurda quantidade de recursos para manuseá-los. Grandes empresas, como a Google, investem fortunas para custear mais datacenters para suportar todo esse volume de dados. Entretanto, não existe limite para o crescimento desses dados, enquanto que o custo para manutenção da infra-estrutura que suporta tudo isso continua a crescer. São investimentos inviáveis para empresas de porte menor.

Investir em datacenters não parecia uma solução inteligente. Pesquisadores investigaram uma nova forma de tratar essa massa de dados, de forma a viabilizar esse serviço, reduzindo custos e aproveitando recursos já existentes. Assim, surgiu a idéia de Ambient Cloud, que utilizaria recursos ociosos de quaisquer dispositivos que pudessem ter acesso à internet. O recurso seria oferecido voluntariamente pelos seus usuários. O grande desafio seria implementar uma forma eficiente de manusear todos esses recursos, de modo a distribuir dados estrategicamente, sem interferir no bom funcionamento do dispositivo do usuário, isto é, consumir excessivamente os recursos disponibilizados ao ponto de atrapalhar a performance do dispositivo.

II. CONCEITO

Muitos conhecimentos são dedicados ao mau uso de recursos. Botnet é um deles. São computadores infectados capazes de se reproduzirem de forma a infectar massivamente através da rede conectada a eles, utilizando os recursos de worms, um malware auto-reprodutor. Seus ataques têm grande alcance, sendo registrados em torno de 700.000 ataques ao

redor do mundo em um período de aproximadamente quatro dias. Na Figura 1 é apresentada uma amostra da captura de ataques feitos por botnets ao redor do mundo. A Clarified Networks gerou uma visualização desses ataques através de logs captados pela F-Secure e recursos disponibilizados pela NASA.

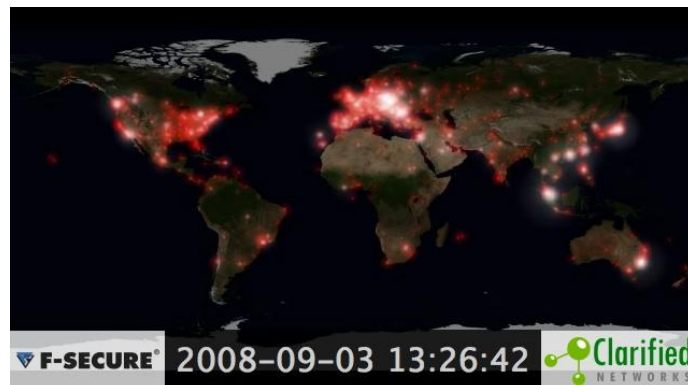


Figura 1 – Amostra da captura de ataques de botnets.

Apesar de ser uma idéia voltada ao mau uso, o conhecimento deixado é muito interessante, possivelmente influenciando no futuro da Cloud Computing.

A palavra “ambient” traz o conceito de envolver completamente. Sendo assim, o conceito “ambient cloud” é a coleção de recursos massivamente distribuídos a partir de qualquer dispositivo que esteja disponível. Esses recursos crescem numa taxa exponencial. Ambient Cloud não é um projeto desenvolvido por alguma organização, muito menos é um projeto de custo estimado em bilhões de dólares, como os datacenters. É uma comunidade, onde cada um contribui com memória, rede, ou qualquer outro recurso que esteja disponível para compartilhar.

III. PROBLEMA

Hoje em dia, o maior problema discutido entre as empresas provedoras de serviços é a quantidade de recursos disponíveis. Mesmo para grandes empresas, assim como a Facebook, o custo para a manutenção de todos os recursos utilizados é caríssimo. Redes Sociais são os maiores consumidores de recursos. Elas precisam de todos os seus dados ativos a maior parte do tempo, devido aos seus acessos. Pessoas do mundo todo estão interconectadas de forma a haver uma quantidade massiva de troca de informações indefinidamente. Utiliza-se muita memória em cache para suportar grandes quantidades

de dados que demandam uma resposta rápida. Para se ter uma idéia, a Facebook possuía, até 2010, mais de 30 mil máquinas, 300 milhões de usuários, 80 bilhões de fotos (sendo que eram requisitadas 600 mil fotos por segundo), 28 terabytes de cache, eram gerados 25 terabytes de logs por dia e requisitados 120 milhões de queries por segundo.

Enquanto empresas gastam rios de dinheiro, a massa de dados continua a crescer. O acesso à internet tornou-se freqüente. Não são mais apenas computadores que podem acessá-los, PDA's, smartphones, celulares, modem 3G, são diversos os pontos de acesso e estão por toda a parte. Dessa forma, se cada habitante do planeta possuir um dispositivo que tenha acesso à internet e o utilize para isso, haverá 7 bilhões de usuários navegando pelos sites. Não há arquitetura que suporte tanta gente. Mas isso não é tudo! Dispositivos conseguem se comunicar de forma autônoma para comunicar informações: tais como sensores de previsão de tempo, sensores inseridos em pessoas, sistemas de monitoramento, etc. Dessa forma, não são apenas aqueles 7 bilhões de acessos. É necessário somar os dispositivos de funcionamento próprio, o que eleva drasticamente esse número de acessos. Para 2020, é estimado que existam em torno de 50 bilhões de dispositivos conectados à internet. O que pesa muito é a velocidade que demandam esses acessos. Ninguém gosta de esperar uma resposta que demore mais de 1 segundo. O interessante é que, nem mesmo as empresas responsáveis pelas redes sociais possuem uma aplicação que englobe recursos do mundo inteiro, mesmo sendo necessário coletar de e fornecer para o mundo todo.

IV. NOVA CULTURA

Quando precisamos de um espaço de armazenamento acessível de qualquer lugar do mundo, não pensamos em outros locais senão os oferecidos pela Google, Microsoft, Facebook, Amazon, etc. Por quê? Porque eles têm fama, são conhecidos e detêm grande confiabilidade. Competir com eles, hoje em dia, está fora de cogitação. Os valores para a alocação de recursos ainda é um investimento baixo com bons resultados. Porém, em um futuro relativamente próximo, recursos vão secar e as taxas vão subir. Nesse nicho entra o Ambient Cloud, onde não será necessário nenhum investimento pesado em infra-estrutura, bastado apenas possuir aplicações capazes de administrar os recursos disponibilizados pelo mundo. Recursos poderão ser trocados, alugados ou vendidos, dependendo do que o fornecedor de recurso desejar ou combinar. É um projeto cooperativo, onde cada pessoa disponibiliza o excesso de recursos que possui para benefício do restante das pessoas ao redor do mundo. A estratégia de distribuir o processamento de dados por diversos dispositivos tem a vantagem de se prevenir de ataques que tentam derrubar o servidor, exatamente por não existir uma central onde são alocados todos os recursos.

As aplicações deverão abranger os recursos oferecidos, tendo a cautela de considerar fatores como: API's, custo, energia, latência, tamanho, perdas, segurança, banda,

localização geográfica, regulamentos, confiabilidade e algoritmos.

V. SEM INTERVENÇÃO HUMANA

Com o desenvolvimento do Ambient Cloud, será possível implementar uma administração da rede, de forma a torná-la cada vez mais eficaz através de métodos que calculam a melhor localização geográfica de seus dados, escolhendo benefícios de custo ou de recursos oferecidos e analisando a confiabilidade através de determinados parâmetros. A aplicação possuirá uma coleção de contratos que permitirá a movimentação de seus dados pelos recursos. Será capaz de personalizar a forma de armazenamento de dados conforme a sua requisição, baseando-se na segurança, latência, preço, credibilidades.

Spanner tenta seguir a mesma política. Desenvolvida pela Google, seu objetivo é prover suporte a 10 milhões de servidores, 10^{13} diretórios, 10^{18} bytes de armazenamento, atendendo até 10^9 clientes. O número é surpreendente. O objetivo é distribuir as execuções computacionais pelo mundo, de forma automática e dinâmica, tentando minimizar ao máximo a latência, o custo da banda de internet, a perda, energia, o uso de recursos computacionais e as falhas.

Sistema automatizado como este só é viável para uma empresa gigante, assim como a Google. É o único sistema, até agora, que aparentemente pode suportar um serviço mundialmente escalável. Entretanto, a Google não pretende utilizá-lo como recursos disponíveis ao mercado, mas como um plano econômico interno.

Apesar de ser um sistema "esperto", não utiliza inteligência artificial em seu algoritmo.

VI. EXCESSO DE RECURSOS

Para 2014 é estimado que existam 2 bilhões de computadores pessoais, recheados com recursos a serem explorados, principalmente poderosos processadores e gigabytes de memória. Também são estimados 7 bilhões de smartphones, que hoje em dia possuem recursos comparáveis a computadores.

Tanto recurso, porém tão pouco explorado. Os dispositivos estão, muitas vezes, isolados ou em ambientes com pouca interação, desperdiçando muito suas capacidades. Ao juntar todas as ociosidades deles, haverá um recurso com capacidade assombrosa.

O problema de recursos disponíveis em celulares está na sua latência. A implementação de aplicativos baseados em baixa latência não é atraída pelo desenvolvimento de uma cloud em cima desses recursos. Porém aplicações com políticas de execução em paralelo e que trabalham com curtas latências podem achar muito interessante a utilização deles. Execuções em paralelo poderão aumentar a eficiência na resposta de requisições, além de poderem consumir menos que outros dispositivos.

Apesar de serem desenvolvidos dispositivos cada vez mais potentes, há um considerável investimento na eficiência energética relativa a eles, onde as baterias duram mais e a recarga demanda menos.

Existem estimativas de que o SmartGrid, formado por sensores de monitoramento (água, clima, tempo, ataques terroristas, tráfego, etc) possa chegar a ser 1000 vezes maior que a Internet em si., o que nos induz dizer que é um forte candidato a ser uma plataforma para o ambient cloud.

O Ambient Cloud não pretende descartar os datacenters. O que já existe permanece, continua a ser utilizado, enquanto outros recursos são adquiridos, aproveitando a ociosidade de recursos. Até 2007 existiam pouco mais de 30 milhões de datacenters.

Body Area Networks também podem entrar como recurso disponível para o Ambient Cloud. Esses pequenos sensores, que são incorporados ao corpo humano, podem se comunicar, podendo transmitir dados a outros dispositivos. Assim como outros dispositivos, possuem recursos ociosos que podem ser aproveitados.

A quantidade de dispositivos que podem ser utilizados para suportar a Ambient Cloud supera e muito a quantidade de dispositivos que são utilizados para suportar a Cloud comum. São trilhões versus dezenas de milhões.

Uma pequena estipulação mostra o quanto é válido adotar essa idéia:

Imagine que você consiga 100 mil voluntários que te ofereçam recursos que dão em média 2 cores, 4 GB de memória RAM e 30 GB de memória em disco por computador. Como resultado, você terá disponível 200 mil cores, 400 TB de memória RAM e 30 PB de memória em disco. Como podem ter observado, todo esse recurso é oferecido voluntariamente, o que implica em zero de custo, enquanto que o custo que a Amazon, por exemplo, teria seria algo exageradamente caro.

VII. ESPECULAÇÕES

Apesar de ainda não existir nenhuma implementação de Ambient Cloud, existem vários estudos. Abaixo alguns possíveis futuros da Ambient Cloud:

A. Compute Grid

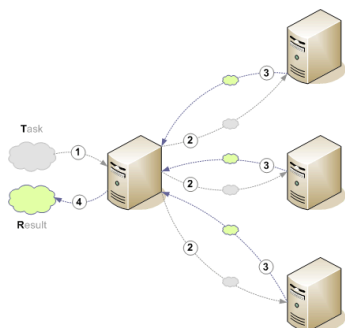


Figura 2 – Compute Grid

Framework capaz de suportar execuções em paralelo entre diversos computadores, não sendo perceptível nenhum decaimento de performance entre eles.

B. Autonomic Computing

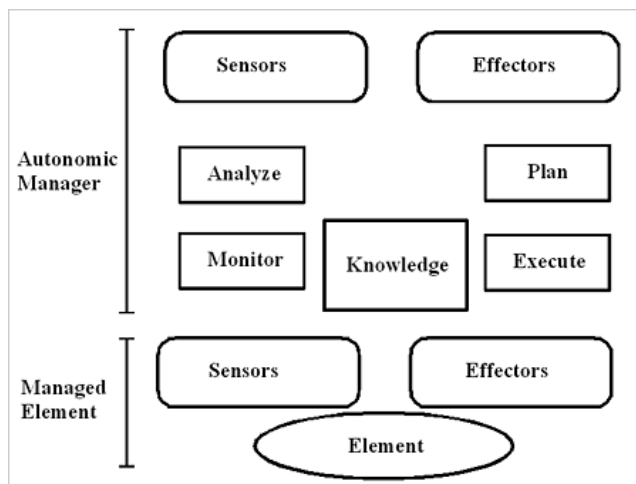


Figura 3 – Autonomic Computing

Sistemas computacionais auto-administrativos, isto é, o sistema configura, corrige, otimiza e protege a si mesmo de forma autônoma. Entretanto, é uma solução que depende de algumas soluções de problemas fornecidas pelo desenvolvedor. Utilizando sua inteligência artificial e seu aprendizado, definirá as próximas soluções.

Não parece ser uma abordagem que terá sucesso, por seguir uma política *top-down*, onde é considerado primeiro o planejamento e depois o adequamento ao mundo real.

C. FAWN (Fast Array of Wimpy Nodes)



Figura 4 – FAWN (Fast Array of Wimpy Nodes)

Arquitetura de cluster rápida, escalável e com eficiência energética. Com ela é possível conectar grande quantidade de pequeninos nós com processadores embutidos e pequena quantidade de memória *flash*. Com isso, é possível atender à requisição de 1300 queries por segundo por nó, enquanto utiliza 4 watts de energia por nó.

Forte candidato a ser implementado para a Ambient Cloud.

D. OceanStore

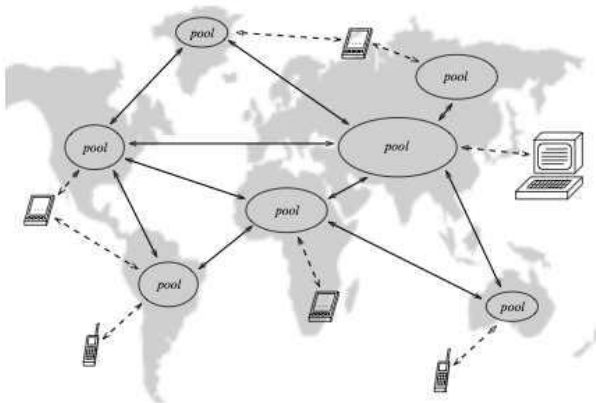


Figura 5 – OceanStore

Infra-estrutura desenhada para se expandir pelo mundo e prover acesso contínuo a dados persistentes. Para proteger seus dados, são utilizadas técnicas de redundância e criptografia. Para boa performance, é permitido o carregamento de dados em cache em qualquer momento ou lugar. O monitoramento previne ataques a serviços além de auxiliar na performance através de movimentos de dados pró-ativos.

E. Alocação de Recursos Baseada no Mercado

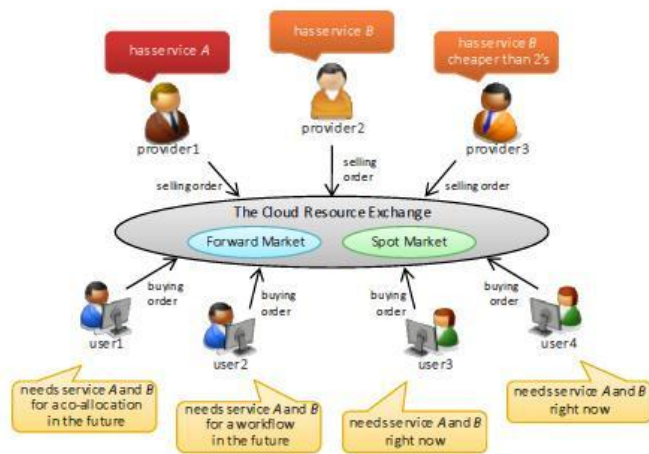


Figura 6 – Alocação de recursos Baseados no Mercado

Existem diversos Recursos não utilizados pelos usuários, então por que não vendê-los? Os recursos que estão “sobrando” podem ser negociados de forma a beneficiar os dois lados, um com a remuneração pela disponibilização de recursos, outro por receber recursos acessíveis remotamente.

O problema é que atualmente não existe um mercado para recursos computacionais. Além disso, as arquiteturas de aplicações não são flexíveis suficientemente de forma a disponibilizar com política voltada à venda de recursos.

VIII. MAPREDUCE

MapReduce é um framework desenvolvido e patentado pela Google, capaz de distribuir tarefas entre vários nós. O MapReduce é dividido, basicamente, em duas funções:

F. Map

A entrada é dividida em vários fragmentos, cada um contendo um par com chave e valor. Essa divisão pode ser executada por diversas máquinas em paralelo. Cada fragmento é chamado arquivo intermediário e estará acessível à função Reduce.

G. Reduce

Tem acesso aos arquivos intermediários, relacionando-os através das chaves contidas neles para possivelmente formar um conjunto menor de valores.

Apesar de ser uma tecnologia inovadora, na prática, acaba não sendo bem aproveitada por não existirem tantos recursos disponíveis ao redor do dado requisitado.

Abaixo, as etapas de um MapReduce em funcionamento:

- 1) A biblioteca de MapReduce inserida ao programa do usuário particiona a entrada em M fragmentos (em geral, 16 MB cada). Geram-se muitas cópias do programa para clusters de máquinas.
- 2) Uma das cópias do programa é especial, o nó-mestre. O restante são nós-trabalhadores que recebem tarefas de Map ou Reduce atribuídas pelo nó-mestre quando o nó-trabalhador está inativo.
- 3) O nó-trabalhador de função Map faz a leitura do fragmento que lhe foi atribuído e passa pela função Map, gerando um arquivo intermediário contendo chave e valor que são armazenados na memória.
- 4) Periodicamente, os arquivos do buffer são armazenados em discos locais, particionados em R regiões. As localização é informada ao nó-mestre, que por sua vez se responsabiliza a encaminhar essa informação aos nós-trabalhadores de função Reduce.
- 5) Quando um nó-trabalhador de função Reduce é informado dessa localização, ele lê os dados situados nessa

localização através de procedimentos de chamada remotos. Ao finalizar a leitura, o nó-trabalhador agrupa todos os arquivos de mesma chave.

- 6) O nó-trabalhador de função Reduce passa à função Reduce a chave e os valores dos arquivos intermediários agrupados. A saída é incrementada à saída final, no local reservado para o fragmento correspondente.
- 7) Quando todas as tarefas estiverem concluídas, o nó-mestre acorda o programa e retorna ao código do usuário.

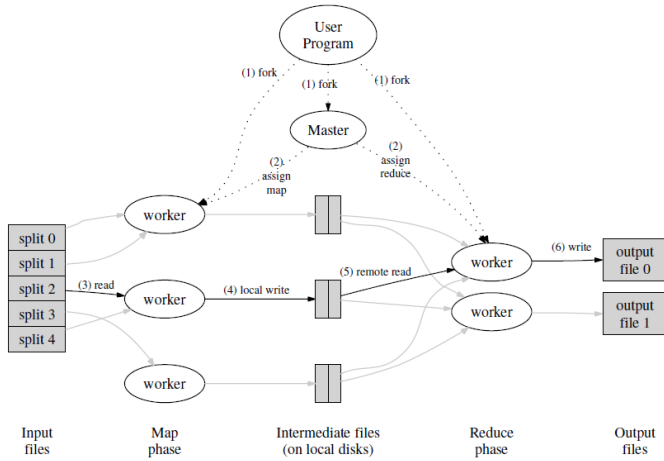


Figura 7 – MapReduce

IX. MISCO

O desenvolvimento de aplicações com sistemas distribuídos voltado a ambientes mobile é de alta complexidade, exigindo a análise de diversos aspectos, tais como concorrência, alocação de recursos, plataforma de desenvolvimento de software e falhas.

Misco é um framework de sistema distribuído desenvolvido para dispositivos móveis. É implementado totalmente em Python e altamente portátil. O desenvolvimento foi baseado no framework do MapReduce, podendo assim otimizar os recursos de dispositivos móveis de uma forma a não desperdiçar capacidade, administrando a baixa conectividade, e disponibilizar serviços ágeis para diversas localizações geográficas.

Na Figura 8, observamos a representação do modelo MapReduce pelo qual a Misco tomou como base.

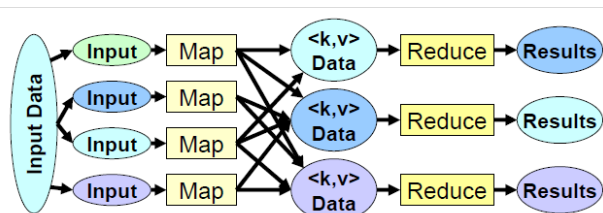


Figura 8 – Modelo Base de MapReduce da Misco

São N aplicações distribuídas executando em M nós-trabalhadores (dispositivos móveis). Cada aplicação distribuída corresponde ao Input. São várias tarefas de Map e Reduce executando em paralelo. Além disso, cada input tem um deadline, isto é, um prazo para ser processado pelo MapReduce. Não existe conexão entre as execuções efetuadas pelos nós-trabalhadores.

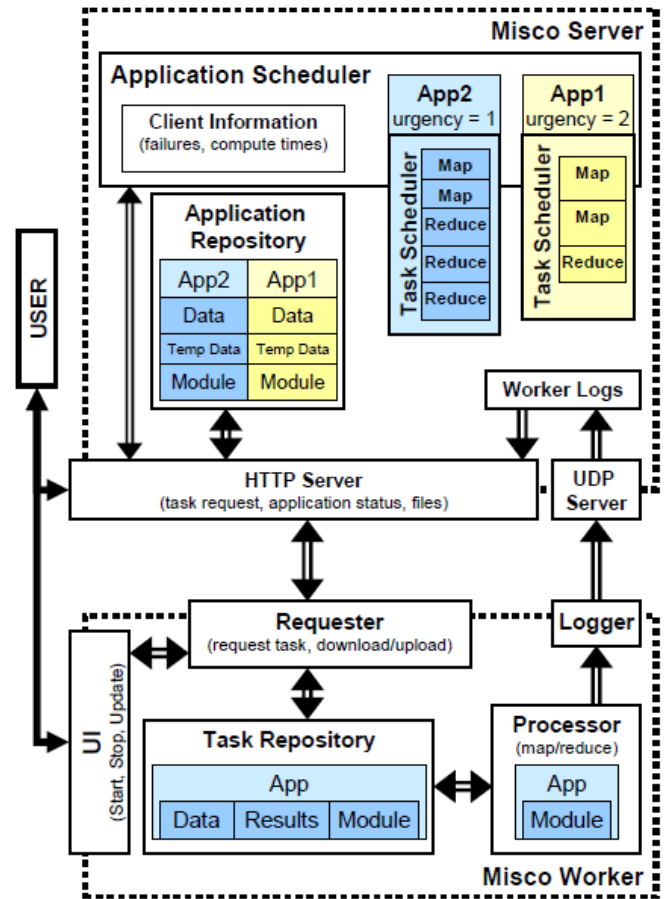


Figura 9 – Arquitetura do Sistema da Misco

Um das mais desafiadoras etapas do desenvolvimento da Misco foi a etapa à qual era necessário definir a forma de implementação do código que permitisse troca de mensagens, recebendo tarefas e transferindo dados. Basicamente, haverias duas formas: sob voto ou por interrupção.

A decisão tomada foi utilizar o sistema sob voto, devido a sua baixa complexidade e custo, além de escapar da necessidade de manter uma conexão entre dispositivos, de modo a garantir que um aparelho receba a tarefa solicitada. Além disso, para implementar interrupções, seriam necessário muitas bibliotecas no dispositivo. Tendo em vista que os dispositivos não são fontes abundantes de recursos, foi descartada rapidamente.

Em sistemas sob votos, a frequência em que ocorrem os votos pode ser configurada, além de poderem selecionar os

servidores que estiverem disponíveis, sempre. Quando não existem tarefas, pode-se tornar inativo por um período e retornar para solicitar tarefas novamente. Porém, se a frequência de eleição for muito alta, haverá um custo alto para manter consultas frequentes sobre novas tarefas. Se for muito baixa, poderá começar a afetar a performance do algoritmo.

Basicamente, o sistema segue a política do Mapreduce, porém existem limitações. Por exemplo, tradicionalmente a implementação teria sistema de arquivos compartilhados, entretanto a realidade impede que tal sistema possa ser implementado. Foi decidido a utilização de HTTP, construído em uma camada confiável de transporte e que já possui suporte através de bibliotecas comuns, para comunicações de requisição, informação de tarefas e transferências de dados.

A principal diferença na implementação de MapReduce na Misco está abaixo:

A. Map

A entrada não possui um formato específico, cabendo ao usuário prover um método que leia e formate o dado para inserir a função Map. São implementados esquemas para reduzir operações, além de nós-trabalhadores serem capazes de particionar seus próprios pares com chave e valor, de forma personalizada. Por fim, gera dados duplicados, podendo também, tratar redundâncias.

B. Reduce

As tarefas de Reduce são muito mais simples. A entrada para esta função é uma lista de pares chave-valor gerada pela função Map. A operação de Reduce será trabalhada em cima dessa lista e o resultado final será gerado. Os resultados são salvos como arquivos e subidos (upload) no servidor.

X. CONCLUSÃO

Assim como a abordagem de NanoDataCenters, a Ambient Cloud ainda não foi realmente implementada abrangendo todos os recursos desejáveis, sendo grande parte das abordagens citadas acima ainda projetos e matéria de estudos de muitos pesquisadores. O conceito segue a mesma linha, tentando utilizar recursos ociosos disponíveis ao redor do mundo.

Tanto NanoDataCenters como Ambient Cloud são a única saída visível no momento, para resolver o problema dos datacenters, que demandam altos investimentos. Caso não sejam implementados, haverá grande problema de disponibilidade de recursos, em breve.

REFERÊNCIAS

- [1] Building Super Scalable Systems: Blade Runner Meets Autonomic Computing In The Ambient Cloud. Disponível em: <http://highscalability.com/blog/2009/12/16/building-super-scalable-systems-blade-runner-meets-autonomic.html>
- [2] Applications Become Black Boxes Using Markets To Scale And Control Costs. Disponível em: <http://highscalability.com/blog/2010/1/17/applications-become-black-boxes-using-markets-to-scale-and-c.html>
- [3] The Amazing Collective Compute Power Of The Ambient Cloud. Disponível em: <http://highscalability.com/blog/2010/2/15/the-amazing-collective-compute-power-of-the-ambient-cloud.html>
- [4] Botnet IRC joins. Disponível em: <https://www.clarifiednetworks.com/Blog/2009-01-01%2018-15>
- [5] Misco: A MapReduce Framework for Mobile Systems. Disponível em: <http://www.cs.ucr.edu/~jdou/misco/>
- [6] MapReduce: Simplified Data Processing on Large Clusters. Google Inc. <http://labs.google.com/papers/mapreduce-osdi04.pdf>