CloudShift: Mudando sua percepção sobre jogos

Nathan de Oliveira Nunes
Orientador: Prof. Dr. Alfredo Goldman
Coorientador: Bruno Motta

Maio, 2023

1 Introdução

1.1 Jogos e tecnologia

Jogar vem se tornando uma atividade cada vez mais comum entre as pessoas. Dados indicam que na ultima década mais de um bilhão de pessoas passaram a jogar com alguma frequência como mostra a figura 1, e que a tendencia no consumo desse tipo de mídia é crescente. Não apenas o numero de pessoas que jogam esta aumentando, como também esta aos poucos se tornando uma atividade mais uniforme entre gênero. A figura 2 mostra que jogar esta se tornando cada vez mais igualitária em termos de gênero, diminuindo consideravelmente a diferença de gosto pela mídia. Outro aspecto importante é a idade media dos jogadores, muitas vezes associada ao publico mais jovem, vem crescendo sua adesão em outras faixa etárias conforme a familiaridade com a mídia cresce 3.

A tecnologia vem evoluindo nas ultimas décadas, tornando computadores, consoles, celulares e outros dispositivos mais poderosos e acessíveis. A miniaturização do hardware tornou possível inovações como os consoles portáteis e jogar em *smartphones*, no entanto a tendencia de miniaturização do hardware vem decrescendo cada vez mais [2]. O fenômeno da tendencia de miniaturização do *hardware* é conhecido como *Moore's Law* ou lei de Moore. A lei é na verdade uma observação feita pelo cofundador da Intel, Gordon Moore, que dizia que o numero de transistores em um chip iria dobrar a cada dois anos.

Ao passo que a tecnologia evoluiu, os jogos também evoluíram. Os primeiros jogos surgiram na década de 1950 e 1960, com jogos como *Bertie the Brain* e *Spacewar!*, e contavam com recursos de *hardware* bem reduzidos e possuíam poucos recursos. Aos poucos, inovações como jogos 3D e gráficos mais realistas foram surgindo, sempre demandando mais do hardware existente.

A medida que a forma da evolução do *hardware* movida principalmente pela miniaturização do transistor começa a desacelerar, novas formas de desenvolvimento da

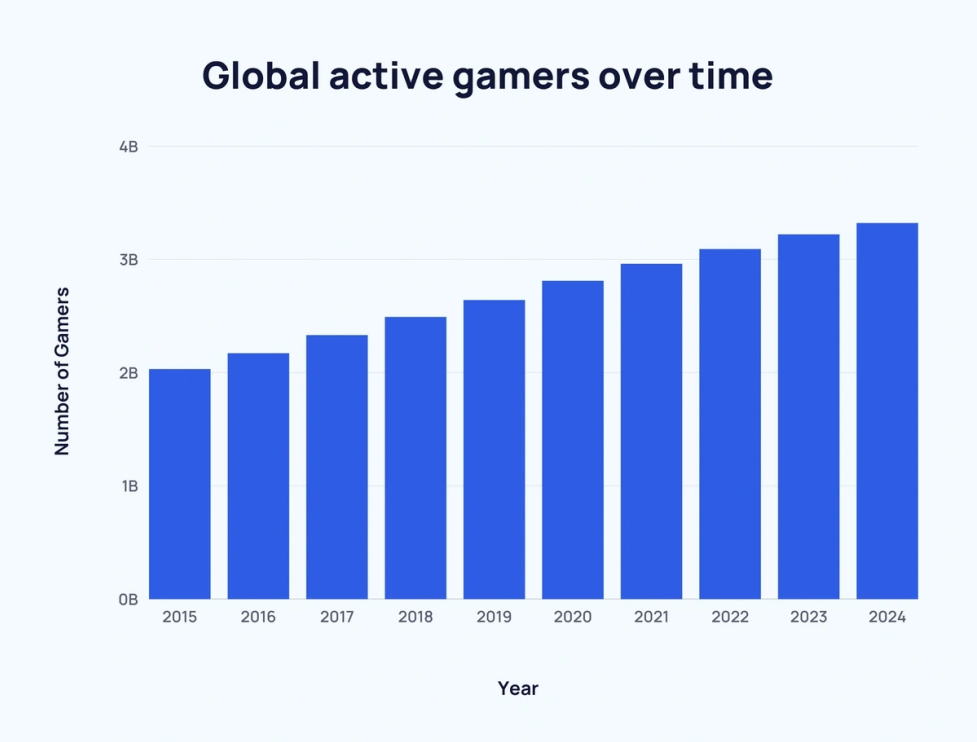

Figura 1: Crescimento do numero total de jogadores ao longo dos anos. Fonte: *Exploding topics*

tecnologia vem sendo propostas como melhorias tanto em software como em arquitetura de hardware, como por exemplo, a especialização do hardware para determinados tipos de tarefas [1]. Um exemplo tradicional desse tipo de especialização é a GPU (*graphics processing unit* que é responsável por processar o pipeline gráfico (projeção, renderização, etc) e é o componente mais fundamental para a execução de jogos modernos. Outro exemplo são as TPUs (*Tensor Processing Unit*) muito usadas para aplicações de inteligencia artificial.

1.2 Cloud gaming

1.2.1 Introdução

Cloud gaming ou *Gaming on demand* é um modo de jogar onde o jogo é executado em uma maquina remota e o video é transmitido pela internet. A primeira tentativa de implementar cloud gaming surgir no começo dos anos 2000, com uma *startup* chamada G-cluster (Game Cluster). O G-cluster foi anunciado na E3 de 2000 e lançado em 2003, mas não obteve muito sucesso. Outra tentativa foi de uma empresa chamada Infinium Labs que apresentou um protótipo em 2004 na E3, mas veio a falir em 2008. Mais recentemente no começo dos anos 2010, serviços como Gaikai e OnLive também foram apresentados em um panorama tecnológico mais maduro, no entanto os serviços não conseguiram se mostrar sustentáveis e suas patentes e marcas foram compradas pela Sony.

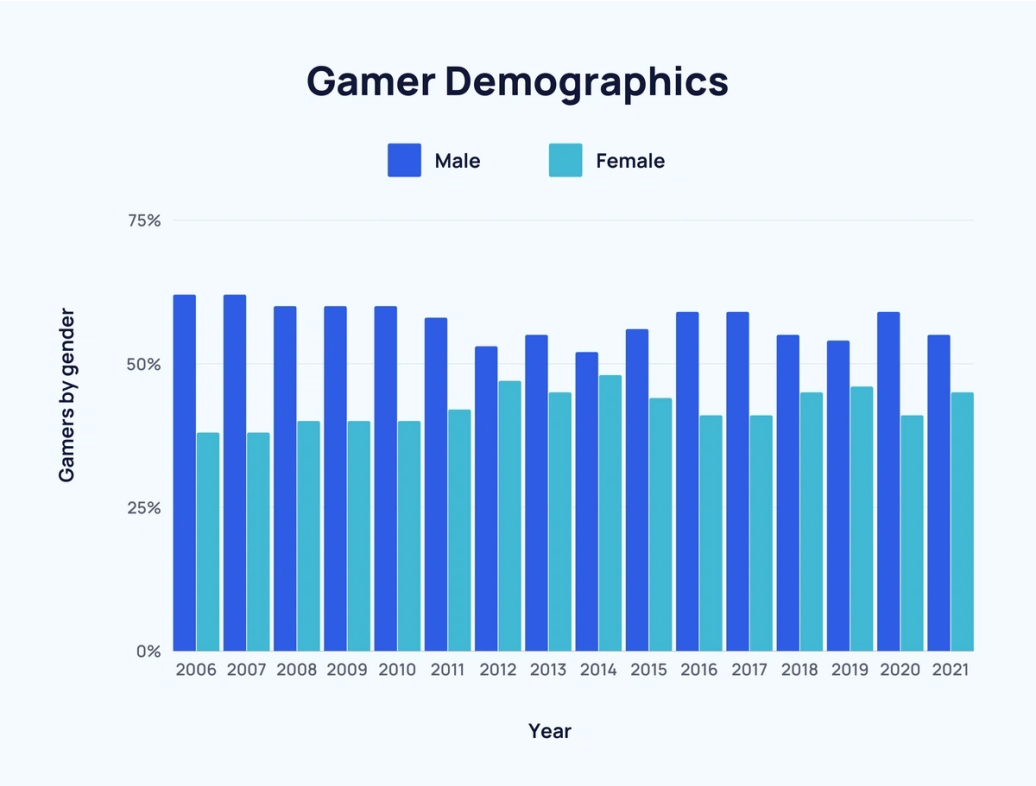

Figura 2: Proporção de gênero ao longo dos anos. Fonte: *Exploding topics*

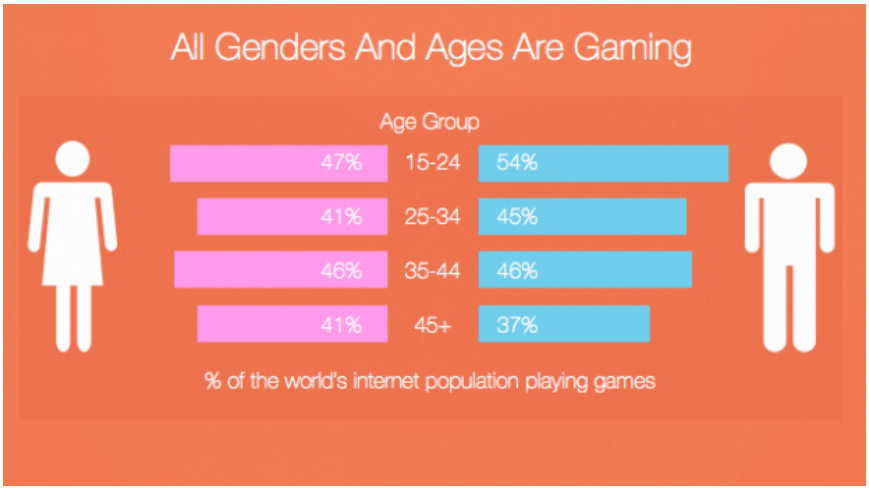

Figura 3: Comparativo da idade dos jogadores. Fonte: *Geekwire*

1.2.2 Principais plataformas

As principais plataformas de Cloud Gaming hoje são dominadas por grandes empresas de tecnologia como Microsoft e NVidia. A Microsoft possui o xCloud, serviço de *cloud gaming* lançado em 15 setembro de 2020. O xCloud é baseado no Xbox, console da própria Microsoft, o qual é utilizado em *datacenters* localizados em varias partes do mundo. O serviço da Microsoft é disponibilizado via assinatura mensal do Xbox Game Pass Ultimate, e possibilita acesso aos jogos disponíveis no serviço sem

custo adicional. Atualmente o serviço possibilita a transmissão em Full HD ate 60 FPS, especialmente pelo fato do serviço ser limitado pelo hardware do Xbox.

Um dos principais concorrentes do xCloud é o Geforce Now da NVidia. A arquitetura da NVidia se baseia em fornecer uma GPU dedicada para cada sessão de jogos, possibilitando gráficos de maior qualidade e mais flexibilidade em questões como suporte a monitores ultrawide e taxas de frame por segundo maiores em relação ao xCloud. O serviço da NVidia também cobra uma assinatura mensal, no entanto você joga através da sua própria biblioteca de jogos adquirida em lojas virtuais como Steam e Epic Games.

O Google também possuía um serviço de *cloud gaming* chamado Stadia, que foi fechado no começo de 2023 pois não obteve sucesso frente aos outros concorrentes.

A Amazon anunciou o Amazon Luna, mas ainda se encontra com baixa disponibilidade.

1.2.3 Desafios

Um dos principais desafios de *cloud gaming* sempre foi a latência. A latência é definida como o tempo de ida e volta de alguma informação na internet. No caso de *cloud gaming* a informação que precisa trafegar na rede e bidirecional, video e som precisam ir pro usuário enquanto a interação do usuário precisa ir pro *datacenter*. Além da latência operacional, relacionada ao processamento das informações pelo jogo, *encoding* e *decoding* de video, temos a latência relacionada a transferência das informações pela internet. A velocidade de transmissão de dados pela internet é limitada pela velocidade da luz, então a qualidade de um serviço esta relacionado com a qualidade da infraestrutura local e sua localização geográfica em relação aos usuários.

Outro desafio esta relacionado as tecnologias envolvidas no processo de execução, renderização e transmissão de dados. Existem muitos *trade-offs* em muitas partes do processo, por exemplo, escolher o nível ideal de compressão de dados, além da qualidade percebida pelo usuário outro fator que aparece nesse cenário é você gastar mais tempo para comprimir um dado ou enviar um dado maior pro usuário. Outra pergunta que surge é como otimizar o seu uso de hardware. Um exemplo disso é o uso da GPU, onde existem serias dificuldades em compartilhar seu uso com outras aplicações simultaneamente [3].

1.3 Principais tecnologias relacionadas

1.3.1 Hardware

Soluções de cloud gaming requerem o uso de hardware especializado e com muita performance pois rodar jogos em escala para milhões de pessoas requer poder computacional que poucas empresas conseguem oferecer. Os principais componentes de hardware em uma solução de *cloud gaming* são a CPU, GPU e memoria RAM. A

depender da arquitetura do sistema, espaço de armazenamento pode não ser crucial considerando que não ha necessidade de armazenar o jogo mais do que a quantidade de instancias em execução. A GPU em particular tende a ser um dos hardwares mais caros e especializados em uma solução de *cloud gaming* pois normalmente requerem o uso dedicado para o serviço, não podendo usualmente serem reaproveitadas ou terem seu poder computacional particionado.

1.3.2 Protocolos

E necessário uma estrategia eficiente para a comunicação de usuário e provedor, uma vez que essa comunicação envolve dados complexos de diferentes tipo e que precisam ter um certo nível de sincronia. Os principais dados envolvidos nessa comunicação é o video, áudio e o controle do jogador. Existem muitos protocolos que transmitem dados dos mais diversos tipos pela internet. Dentre os mais promissores ou já utilizados para fins similares podemos destacar alguns. Um dos mais antigos protocolos para uso de um computador remoto surgiu no começo de 1998 com o protocolo aberto RFB (*remote framebuffer*). RFB suporta diferentes tipos de *encodings* o que permite certa flexibilidade. O protocolo foi utilizado pelos sistemas de VNC (*virtual networking computing*). Os sistemas de VNC foram desenvolvidos no começo dos anos 2000 em um laboratório da AT&T. Esses sistemas funcionam como uma arquitetura cliente e servidor, onde o servidor transmite a tela para o cliente e o cliente interage normalmente com a tela como se fosse um sistema local. Na maioria das implementações de VNC (TigerVNC, TurboVNC, etc) a tela pode ser virtual, ou seja, não ha a necessidade de ter um monitor conectado do lado do servidor, nesse caso dizemos que o servidor é *"headless"*.

Apesar da popularidade dos VNCs para uma grande gama de atividades, seu uso não foi projetado para aplicações que requerem processamento gráfico mais intenso ou mesmo com uma baixa latência suficiente para se jogar jogos mais movimentados e que requerem um tempo de resposta mais rápido como jogos de ritmo ou de tiro. Muitas das implementações de VNC não são aceleradas por GPU com exceção do TurboVNC, o que gera serias dificuldades para se assistir vídeos por exemplo sem ter a sensação de travamento.

Devido a essas limitações outras abordagens começaram a serem testadas, principalmente separando as responsabilidades de encoding, decoding do protocolo de transmissão. A chegada de protocolos como HTTP/2, gRPC e webRTC permitiram que os streaming se tornasse mais simples e com maior performance em relação aos protocolos web anteriores. Combinar ferramentas dedicadas a tarefas especificas e protocolos de transmissão eficientes é uma forma de tornar o problema de *cloud gaming* mais modular e tratável.

Outro protocolo que merece atenção é um protocolo criado pela NVidia especialmente para jogos chamado GameStream, esse protocolo foi criado inicialmente para streaming utilizando os dispositivos NVidia Shield. O suporte da NVidia ao protocolo foi descontinuado no começo de 2023, mas existem implementações open source

do protocolo.

1.3.3 Ambientes

Para tornar uma solução de *cloud gaming* viável é necessário que os sistema além de ter uma boa qualidade consiga ser escalável. Por exemplo ter a necessidade de um monitor para cada pessoa que esteja usando o serviço elevaria os custos do datacenter de forma que o serviço poderia ser inviável. Tecnologias de virtualização como maquinas virtuais vem sendo desenvolvidas desde o século passado e obtendo sucesso numa alta gama de aplicações. Uma maquina virtual permite características desejáveis como isolamento e capacidades mais flexíveis em relação a uma maquina física. Soluções de virtualização como Virtual Box e KVM permitem a passagem de uma GPU para uma maquina virtual através de uma tecnologia conhecida como *GPU passthrough*, no entanto essa solução não permite o compartilhamento da GPU com mais de uma maquina virtual.

Containers vem surgindo como alternativas viáveis a maquinas virtuais por serem mais leves. *Containers* são capazes de isolar a execução de aplicações ao mesmo que compartilham o kernel do host e não precisam de recursos dedicados e fixos pra eles. Isso permite maior flexibilidade na hora de alocar recursos computacionais pra os containers conforme a necessidade além da maior facilidade de manutenção. Docker é o principal tipo de container, onde a configuração do container é especificada através de um arquivo chamado Dockerfile. Algumas placas da NVidia possuem a capacidade de serem usados por mais de um container ao mesmo tempo. Essa demanda por compartilhar GPUs entre diferentes containers surgiu principalmente por conta do uso GPUs para treinamentos de modelos de *machine learning*.

Kubernetes é um orquestrador de containers de código aberto desenvolvido pelo google, sendo um dos maiores projetos open source do mundo junto com o kernel do Linux e o banco de dados PostgreSQL. Kubernetes funciona abstraindo o hardware em *"resources"* que podem ser requisitados pelas aplicações de diversas formas. Kubernetes possui diversos componentes que precisam ser instalados nos nos que serão parte do cluster, porem os grandes provedores de cloud possuem serviços prontos de Kubernetes onde você não precisa administrar as maquinas onde os componentes do Kubernetes roda. Kubernetes se destaca em aplicações de cloud pois possui muitos mecanismos que fazem com que características desejáveis sejam alcançadas como resiliência e alta disponibilidade ao mesmo tempo que é altamente flexível e consegue ser usado para implementar varias arquiteturas diferentes.

1.3.4 Softwares

Alguns softwares possuem um papel bastante fundamental em *cloud gaming*. Dentre estes podemos destacar o FFmpeg, que é um projeto *open source* que consiste num conjunto de programas e bibliotecas que lidam com video e áudio de diversas formas incluindo para transmissão de dados através de diversos protocolos e formatos.

Outros softwares mais diretamente ligados com jogos que são importantes de serem mencionados são o Parsec e o Moonlight, estes softwares foram feitos pensando em *streaming* e Co-op de jogos utilizando a sua própria maquina pessoal para fazer o hosteamento. O parsec é um software proprietário de uso gratuito e o Moonlight e de código aberto e implementa o protocolo GameStream da NVidia.

2 Projeto

2.1 Contexto

Uma das principais vantagens que *cloud gaming* oferece é a capacidade de economizar recursos de hardware e fazer com que os usuários não precisem dar a manutenção em seu próprio hardware. Também não é necessário desembolsar grandes valores em uma GPU que ficara ociosa a maior parte do tempo, pois o usuário muitas vezes joga apenas nos finais de semana e utiliza o computador para atividades que não precisam de uma GPU para serem concretizadas. O melhor uso do hardware poderia abrir portas tanto para os consumidores que poderiam economizar no hardware quanto para os distribuidores do serviço e criadores de jogos, pois a especialização do hardware e outros recursos semelhantes podem ser viáveis em larga escala.

Ainda ha muitas limitações nos serviços de *cloud gaming* disponíveis atualmente. Serviços como xCloud que usam o hardware do Xbox estão limitados as capacidades do console, enquanto serviços como o Geforce Now que são mais flexíveis, ainda não conseguem aproveitar o uso da GPU para alimentar mais de um jogo ao mesmo tempo.

Existem algumas soluções abertas de *cloud gaming* ou que se propõem a resolver problemas semelhantes. Um exemplo de *cloud gaming* é o Cloud Morph (https://github.com/giongto35/c morph) que implementa uma solução completa de *cloud gaming* mas possui altas limitações devido ao uso do Xvfb, que é um *framebuffer* virtual do X11, o *display server* mais tradicional do Linux. O uso do Xvfb é justificado por ser uma maneira de ter um servidor *headless*, no entanto, o Xvfb não tem suporte a aceleração por GPU o que dificulta a execução de jogos mais modernos.

O Game on Whales é uma proposta que não se propõe exatamente a ser um serviço de *cloud gaming*, mas possibilita que você rode uma ou mais aplicações gráficas em sua maquina e acesse elas remotamente. Diferentemente do Cloud Morph, a arquitetura do Game on Whales utiliza Docker e Sunshine. O Sunshine é uma implementação do lado do servidor do protocolo GameStream da NVidia, sendo então um projeto ligado ao Moonlight. A ideia do Game on Whales é que ele cuide da parte do servidor e o usuário utilize o Moonlight como cliente. Dentro do container, o Game on Whales utiliza uma sessão completa do X11, de forma que não tem a desvantagem da falta de aceleração gráfica do Xvfb.

2.2 Objetivos

O principal objetivo deste projeto é estudar e implementar uma solução ou soluces que contribuam para a área de *cloud gaming* ao resolver problemas chave, principalmente os problemas ligados a performance e escalabilidade desses sistemas.

A figura 4 mostra uma arquitetura de alto nível simplificada de *cloud gaming*. O ponto chave nessa arquitetura é como garantir a adequada transmissão de dados em

condições favoráveis a jogos (qualidade gráfica, baixa latência, etc).

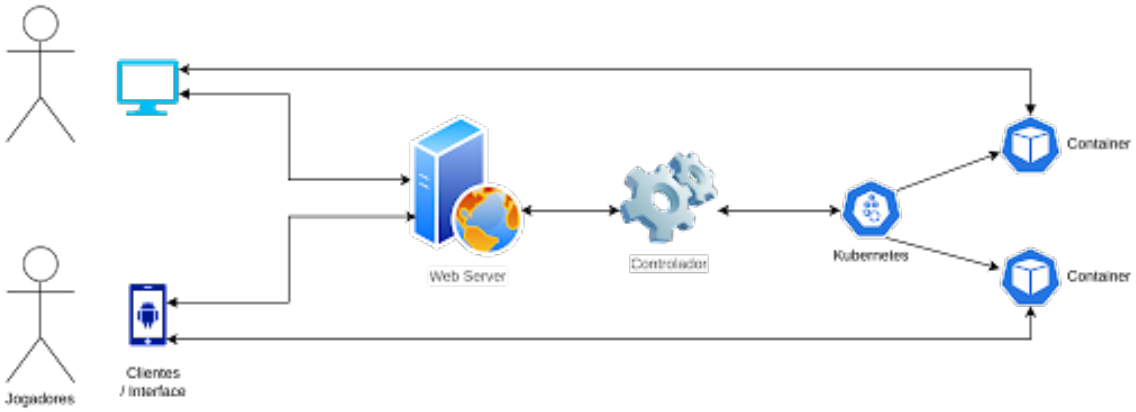

Figura 4: Ilustração de uma possível arquitetura de *cloud gaming*

Como containers são fundamentais para se escalar a maior parte das aplacações atualmente, o foco inicial sera em conseguir atingir um alto nível de performance dentro de um container e depois buscar formas de escalar esses containers.

3 Cronograma

- A1 - Revisar a literatura sobre *cloud gaming*, incluindo suas tecnologias de base e o estado atual da arte.
- A2 - Identificar as principais limitações atuais na área e procurar tecnologias, ferramentas e métodos que possam ser uteis para problemas atuais.
- A3 - Explorar, utilizar e implementar os principais métodos de *cloud gaming* visando contribuir melhorias e montar testes de ponta a ponta.
- A4 - Implementar um protótipo de *cloud gaming* que seja funcional e possua alguma melhoria em algum aspecto em relação as tecnologias atuais.
- A5 - Redigir a monografia final.
- A6 - Preparar a apresentação do trabalho.

	1° Semestre/2022				2° Semestre/2022					
	Março	Abril	Maio	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro
A1	X	X								
A2		X	X							
A3			X	X	X					
A4					X	X	X	X	X	
A5									X	X
A6									X	X

Tabela 1: Cronograma Anual. Detalhamento das atividades ao longo do ano.

Referências

[1] Charles E Leiserson, Neil C Thompson, Joel S Emer, Bradley C Kuszmaul, Butler W Lampson, Daniel Sanchez, and Tao B Schardl. There's plenty of room at the top: What will drive computer performance after moore's law? *Science*, 368(6495):eaam9744, 2020.

[2] John Shalf. The future of computing beyond moore's law. *Philosophical Transactions of the Royal Society A*, 378(2166):20190061, 2020.

[3] Himanshu Yadav and B Annappa. Adaptive gpu resource scheduling on virtualized servers in cloud gaming. In *2017 Conference on Information and Communication Technology (CICT)*, pages 1–6, 2017.