

ANÁLISE E DESEMPENHO DE CONTAINERS

SEMINÁRIO - MAD

Informações Básicas

Nome

Performance analysis of Linux containers for high performance computing applications

Autores

David Beserra, Edward David Moreno, Patrícia Takako Endo, Jymmy Barreto, Stênio Fernandes, Djamel Sadok.

Publicação

International Journal of Grid and Utility Computing (IJGUC), Vol. 8, No. 4, 2017

DOI

[10.1504/IJGUC.2017.10009368](https://doi.org/10.1504/IJGUC.2017.10009368)





Tópicos

- **Introdução**
 - Visão geral sobre HPC, virtualização em nuvem e o problema da sobrecarga.
- **Virtualização em Segundo Plano**
 - Conceitos básicos e comparação entre os tipos de virtualização utilizados.
- **Trabalhos Relacionados**
 - Principais estudos anteriores e as lacunas que este trabalho busca preencher.
- **Design Experimental**
 - Metodologia, infraestrutura e ferramentas usadas nos testes comparativos.
- **Resultados**
 - Análise do desempenho de CPU, comunicação e impacto do compartilhamento de recursos.
- **Conclusão e Trabalhos Futuros**
 - Principais descobertas e próximos passos para ampliar a pesquisa.



PUC Minas

INTRODUÇÃO

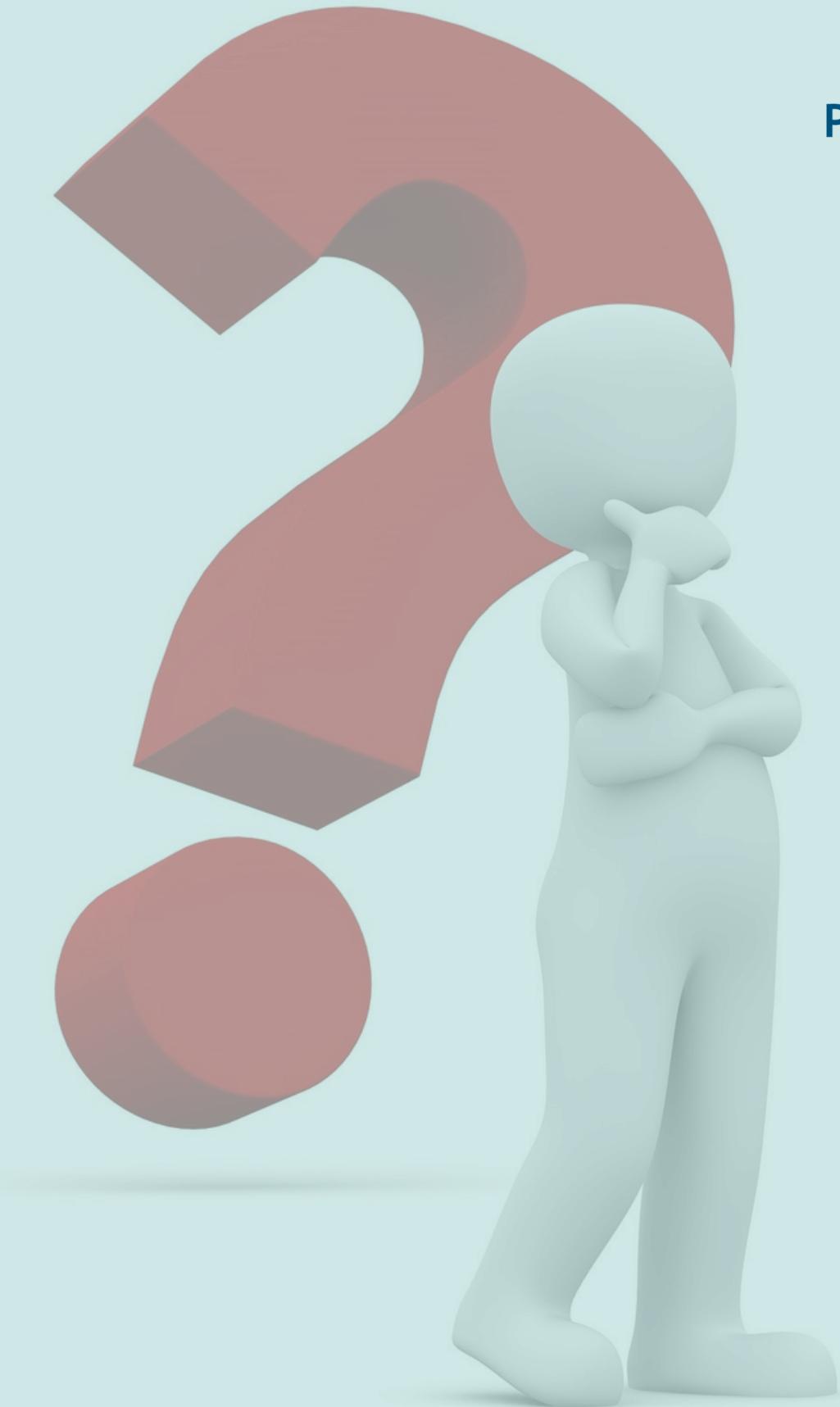


Contexto e Motivação

- A Computação de Alto Desempenho (HPC) é essencial para aplicações intensivas em cálculo e dados, como simulações científicas, modelagem e renderização 3D.
- Tradicionalmente, essas aplicações são executadas em clusters dedicados, com infraestrutura física própria.
- A computação em nuvem surgiu como alternativa flexível, com vantagens como escalabilidade, elasticidade e redução de custos operacionais.
- No entanto, o uso de hipervisores (como KVM) em ambientes virtuais introduz sobrecargas que afetam o desempenho de aplicações HPC.

Problema e Objetivo

- O principal desafio está na sobrecarga causada pela virtualização baseada em hipervisores, que compromete a performance esperada em HPC.
- Containers Linux (como o LXC) surgem como alternativa promissora, oferecendo virtualização leve com desempenho próximo ao nativo.
- **Objetivo do trabalho:**
 - Avaliar comparativamente o desempenho do LXC (virtualização por container) e do KVM (hipervisor) em ambientes de HPC, com foco em CPU e comunicação entre processos.
- A proposta é analisar diferentes cenários e fornecer evidências sobre a viabilidade do uso de containers em HPC.





PUC Minas

VIRTUALIZAÇÃO EM SEGUNDO PLANO

O que é virtualização?

- Tecnologia que permite a criação de múltiplas intâncias isoladas (Maquina virtual ou contêineres) em um único hardware físico.
- O artigo explora como diferentes abordagens de virtualização impactam o desempenho, especialmente em ambiente de Computação de Alto Desempenho (HPC).

Principais abordagens

Virtualização Completa (Full Virtualization):

- Permite que um sistema operacional convidado (guest OS) funcione sem modificações em seu kernel.
- Impõe uma sobrecarga de desempenho significativa na máquina virtual (VM).

Paravirtualização (Paravirtualization)

- Os hardware são acessados através de drivers paravirtualizados especiais, resultando em melhor desempenho em comparação com a virtualização completa.
- Exige modificação no kernel para suportar esses drivers especiais.
- Reduz o consumo de CPU, mas reduz a segurança e aumenta a dificuldade de gerenciamento.

Virtualização Assistida por Hardware (Hardware-Assisted Virtualization):

- Utiliza instruções específicas para virtualização. Tem o objetivo de mitigar sobrecarga de desempenho que corre na virtualização completa.

Virtualização em Nível de Sistema Operacional (OS-level Virtualization / Contêineres)

- Contêineres que permitem que os processos tenham seus próprios recursos de hardware de forma direta.
- Cada contêiner executa sua própria cópia do sistema operacional e sistema de arquivos. Kernel comum a todos contêineres.
- Metaforicamente cada contêiner é uma “Fatia” do sistema operacional tendo sua própria pilha de rede e recursos computacionais.

Objetivo

O principal objetivo do trabalho apresentado no artigo é analisar o desempenho da virtualização em nível de SO em comparação com outra técnica de virtualização.

Soluções de Virtualização

| **KVM (Kernel-based virtual Machine)**

- Solução de código aberto integrada ao kernel do Linux.
- No KVM, o gerenciamento de Entrada/Saída e rede é realizado por uma versão modificada do QEMU.
- As requisições de E/S feitas por uma VM são encaminhadas para o QEMU, que as redireciona para o sistema operacional hospedeiro.

| **LXC (Linux Containers)**

- Solução que não requer emulação de hardware físico.
- O principal uso do LXC é executar uma cópia completa de um sistema operacional Linux dentro de um contêiner, sem a sobrecarga de executar um hipervisor de nível 2.



PUC Minas

TRABALHOS RELACIONADOS



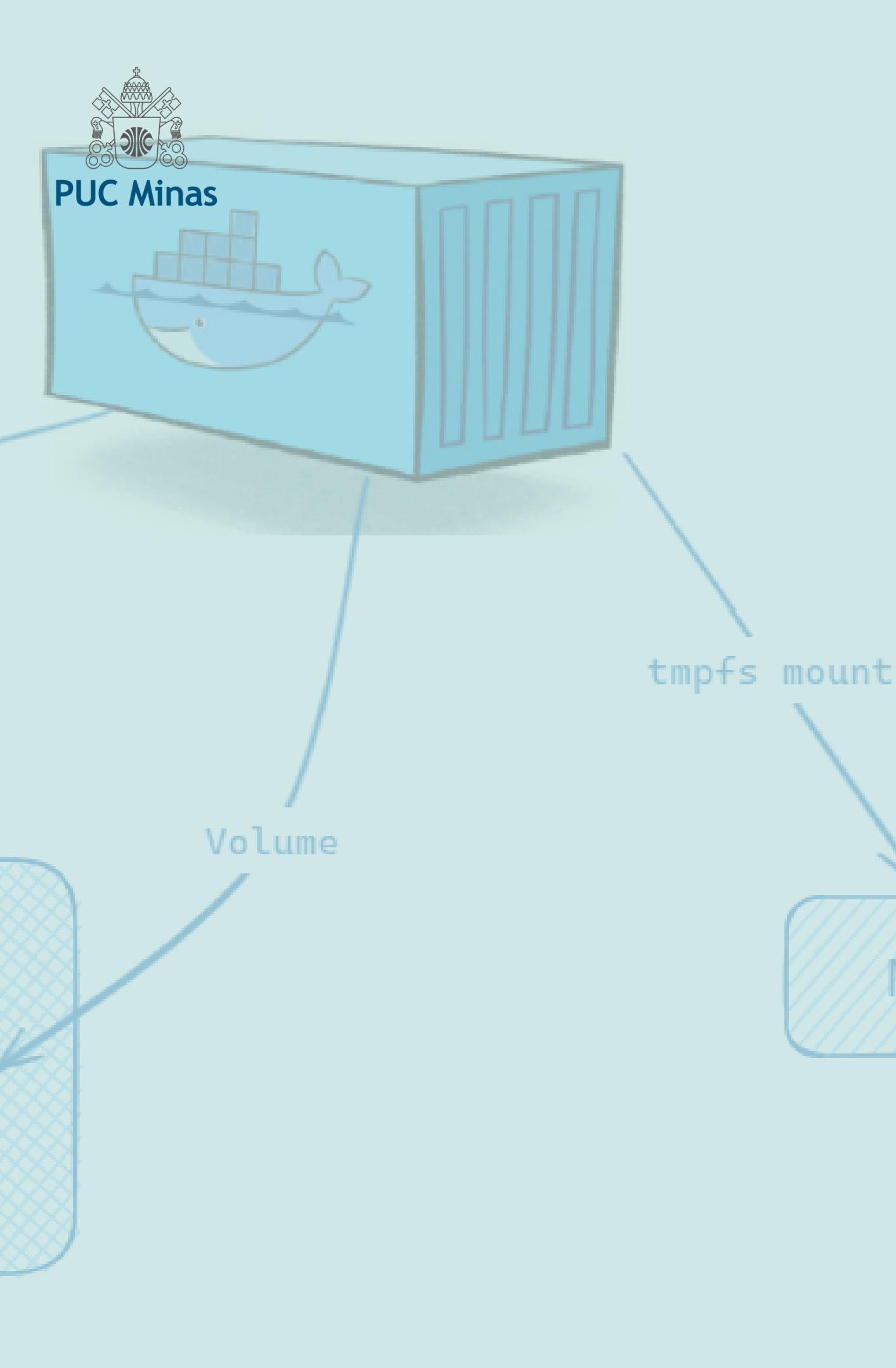
O que outros estudos já mostraram?

- **Nussbaum et al. (2009):**
 - Compararam KVM e Xen, analisando CPU e rede; KVM teve melhor uso do CPU do host em cenários de alta carga de rede.
- **Younge et al. (2011):**
 - KVM e VirtualBox mostraram melhor desempenho global em clusters virtuais; KVM se destacou em capacidade computacional e estabilidade1.
- Esse trabalho focou em **hipervisores, não em containers**.

Por que comparar virtualização em HPC?

- **HPC tradicional:**
 - Uso de clusters dedicados, recursos locais e configuração estática.
- Cloud Computing trouxe elasticidade e facilidade de gestão, mas virtualização pode causar perdas de desempenho.
- Containers, como LXC, surgem prometendo desempenho próximo ao nativo, com menos overhead que hipervisores tradicionais.



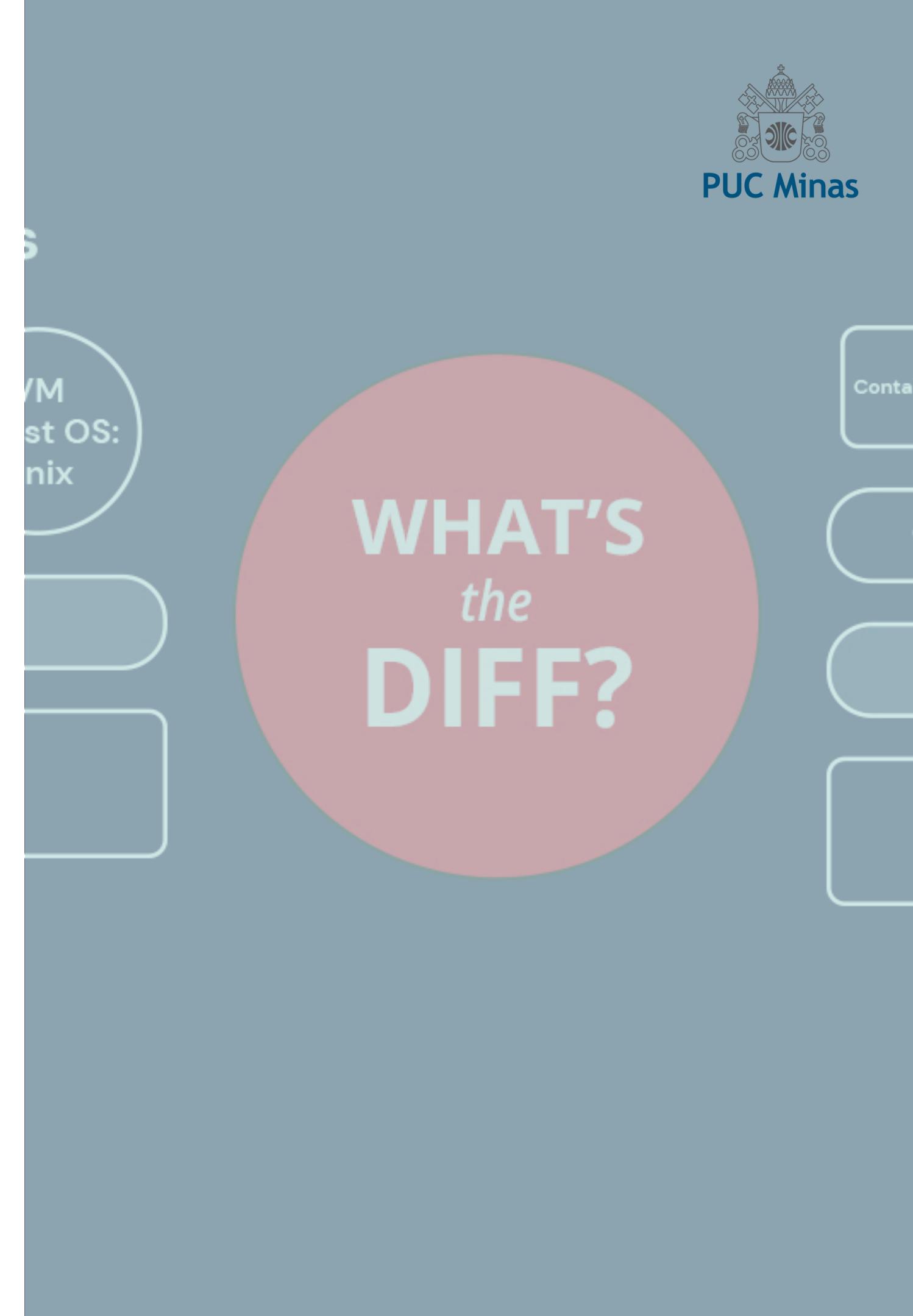


Containers em Ambientes HPC

- **Containers têm vantagens:**
 - Inicialização até 11 vezes mais rápida e uso de disco muito menor que VMs (Seo et al., 2014).
- **Xavier et al. (2013):**
 - Compararam OpenVZ, LXC e VServer com Xen; containers e Xen tiveram desempenho de processamento próximo ao nativo.
- **LXC e VServer se destacaram em rede:**
 - Maior banda e menor latência, tornando LXC o mais indicado para HPC entre containers.

Containers vs Hipervisores

- **Felter et al. (2014):**
 - Docker (container) vs KVM;
 - Docker melhor ou igual que KVM em CPU, memória, armazenamento e rede.
- Ambos têm overheads pequenos para CPU e memória, mas containers escalam melhor e são mais eficientes em ambientes HPC.
- Containers facilitam disponibilidade e uso eficiente dos recursos.





O que o artigo traz de novo?

- Poucos estudos anteriores analisaram containers em HPC de forma detalhada.
- O artigo compara LXC, KVM e ambiente nativo em diferentes cenários de uso e comunicação de recursos.
- **Inovação:**
 - Análise estatística detalhada e avaliação da comunicação intra-nó, ampliando o entendimento sobre virtualização eficiente para HPC.



PUC Minas

DESIGN EXPERIMENTAL

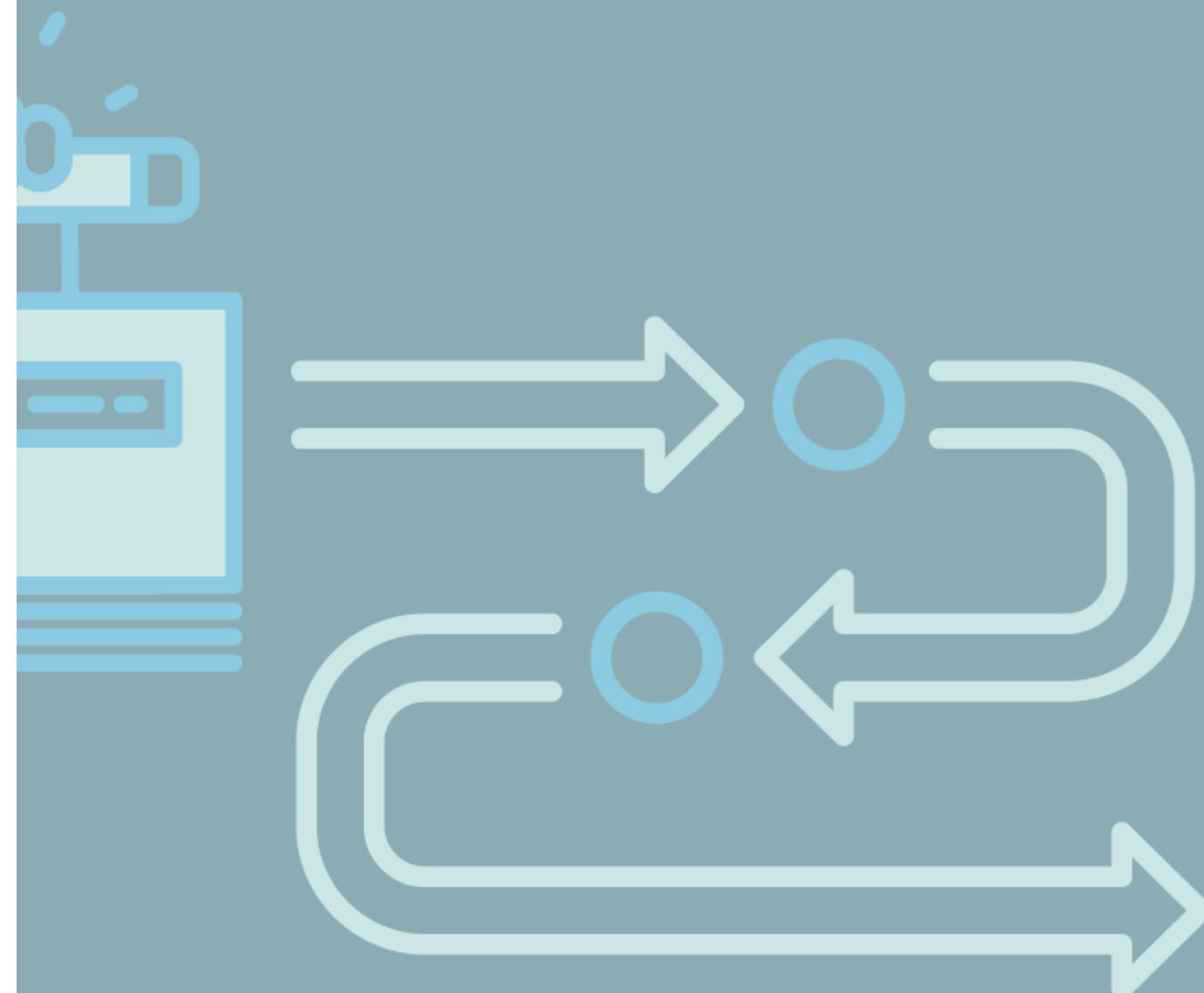
LXC VS KVM

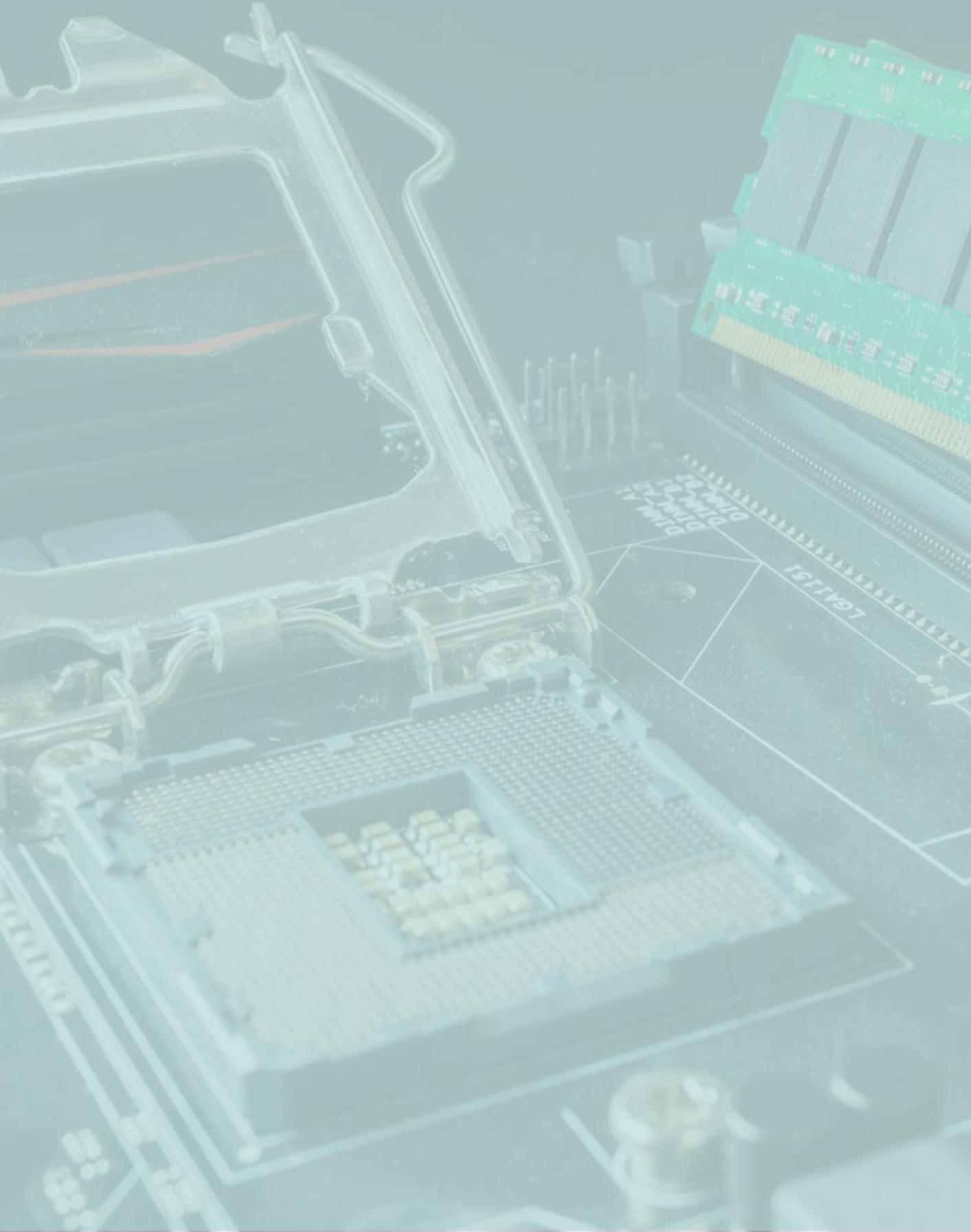
Avaliação: KVM **vs** LXC

- **Avaliação experimental de virtualização:**
 - Comparação entre KVM (hipervisor) e LXC (containers) com ambiente nativo, usando benchmarks e cenários de HPC
- **Objetivo:**
 - Analisar como cada tecnologia impacta o desempenho de aplicações científicas e de alto desempenho, considerando CPU, memória e comunicação
- **Importância:**
 - A escolha entre KVM e LXC pode afetar diretamente a eficiência, o custo e o desempenho em ambientes de nuvem e HPC, sendo fundamental para decisões de infraestrutura

Metodologia

1. Planejamento dos testes
2. Definição das métricas
3. Execução dos experimentos.
4. Tratamento estatístico dos dados, eliminando outliers.
5. Análise dos resultados.





Ferramentas de Teste

- Dois computadores HP EliteDesk 800 G1, ambos com Intel Core i7-4770 (3,4 GHz) e 8 GB DDR3
- Conectados via rede Gigabit Ethernet
- Ubuntu 14.04.2 LTS 64 bits em todos os ambientes
- Ambiente padronizado para isolar o efeito da virtualização



Ferramentas e Métricas

- Desempenho de CPU avaliado com o benchmark HPL (High Performance Linpack)
 - Mede operações de ponto flutuante por segundo (Flops).
- Desempenho de comunicação avaliado com NetPIPE.
 - Mede largura de banda e latência entre processos.

Métricas fundamentais para aplicações HPC, que dependem de eficiência computacional e de comunicação.





Cenários Avaliado

- 4 cenários: nativo, 1 instância, 2 instâncias e 4 instâncias.
- Memória ajustada proporcionalmente em cada cenário.
- Testes realizados sem comunicação entre processos e com comunicação cooperativa.

Quadro de Testes

Table 1 Single-server environments

<i>Environment</i>	<i>Nodes</i>	<i>Cores/node</i>	<i>Memory/node (MB)</i>
Native	1	4	3096
Virtual-1	1	4	3096
Virtual-2	2	2	1536
Virtual-4	4	1	768

Procedimentos Realizados

- HPL executado em modos independente e cooperativo para cada núcleo disponível
- NetPIPE utilizado para medir comunicação intra-nó e entre servidores
- Cada teste repetido 32 vezes para garantir validade estatística
- Descarte dos 2 maiores e 2 menores valores (eliminação de outliers)
- Cálculo de média e desvio padrão com as 30 amostras restantes

Objetivos do Design Experimental

- Avaliar o impacto real da virtualização no desempenho de aplicações HPC
- Analisar efeitos do compartilhamento de recursos na qualidade do serviço
- Auxiliar na escolha da tecnologia mais adequada para HPC e nuvem
- Equilibrar eficiência, custo e desempenho conforme as necessidades do ambiente





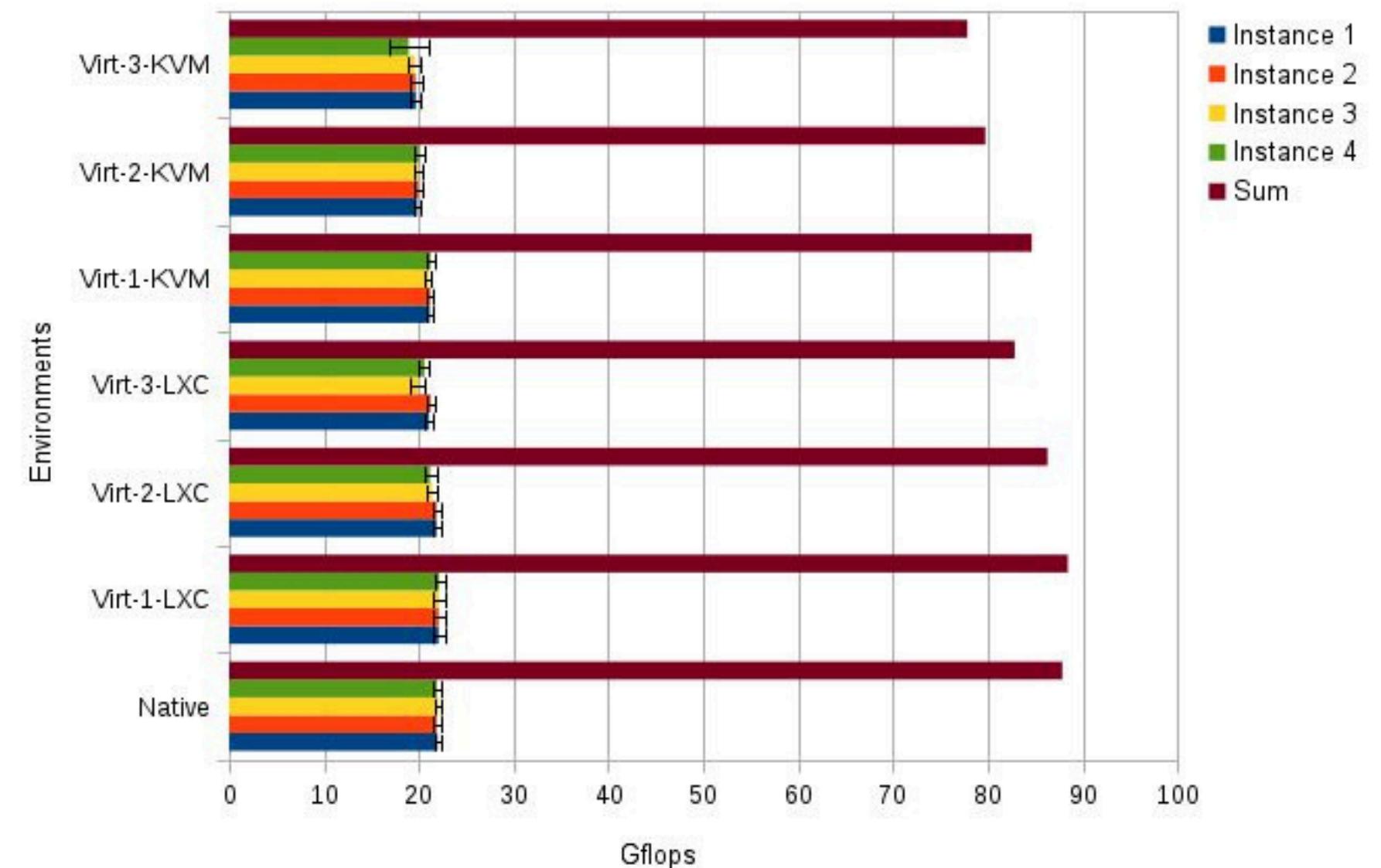
PUC Minas

RESULTADOS

Benchmark de CPU usando HPL

Benchmark de CPU usando o HPL, sem comunicação entre processos.

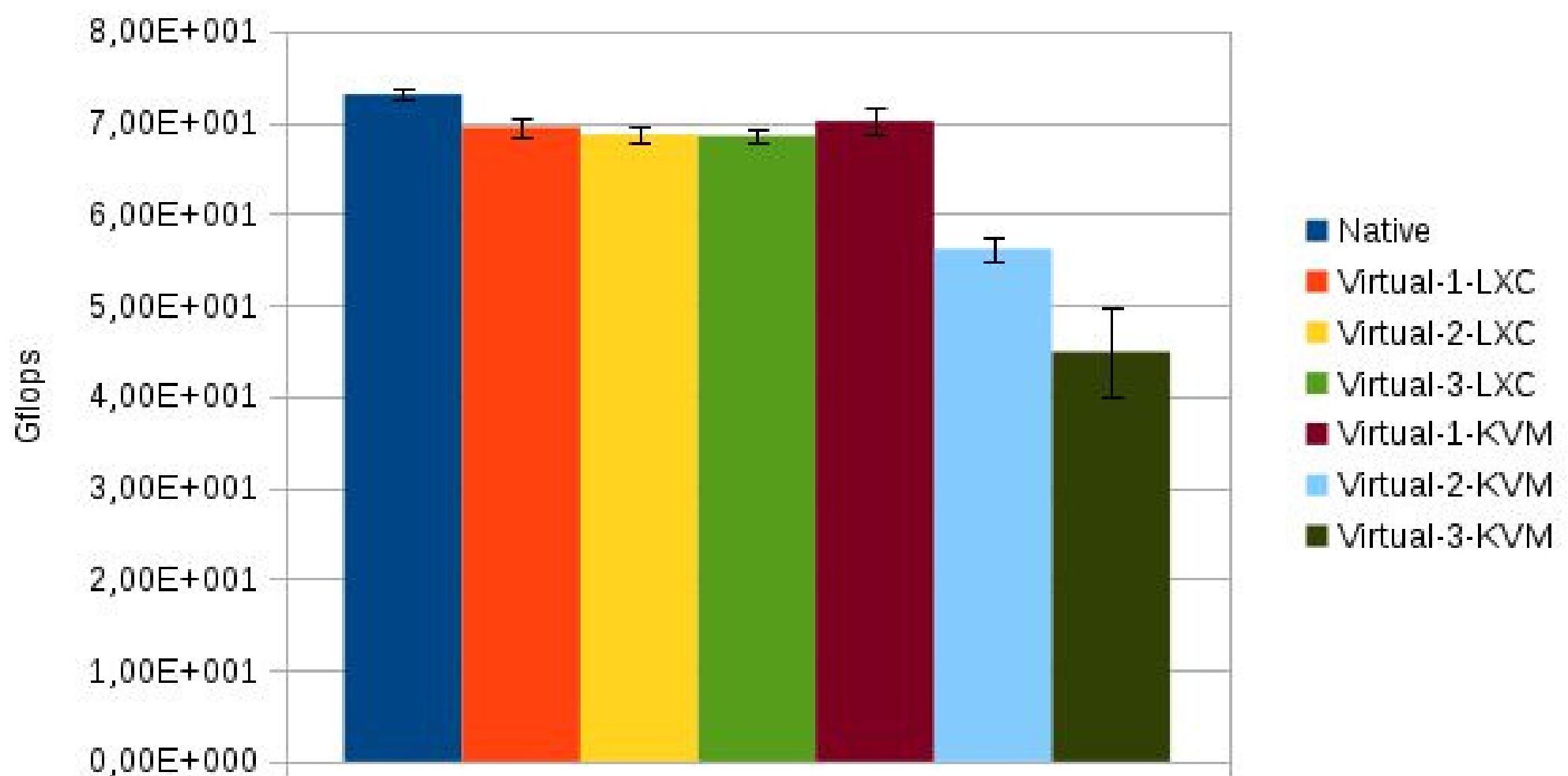
- LXC teve desempenho quase igual ao nativo, mesmo com todos os recursos sendo usados.
- KVM teve desempenho inferior, especialmente em cenários com múltiplas VMs.
- Quanto mais VMs (Virtual-2, Virtual-4), mais o desempenho cai no KVM.



Benchmark de CPU usando HPL

Benchmark de CPU usando o HPL, sem comunicação entre processos.

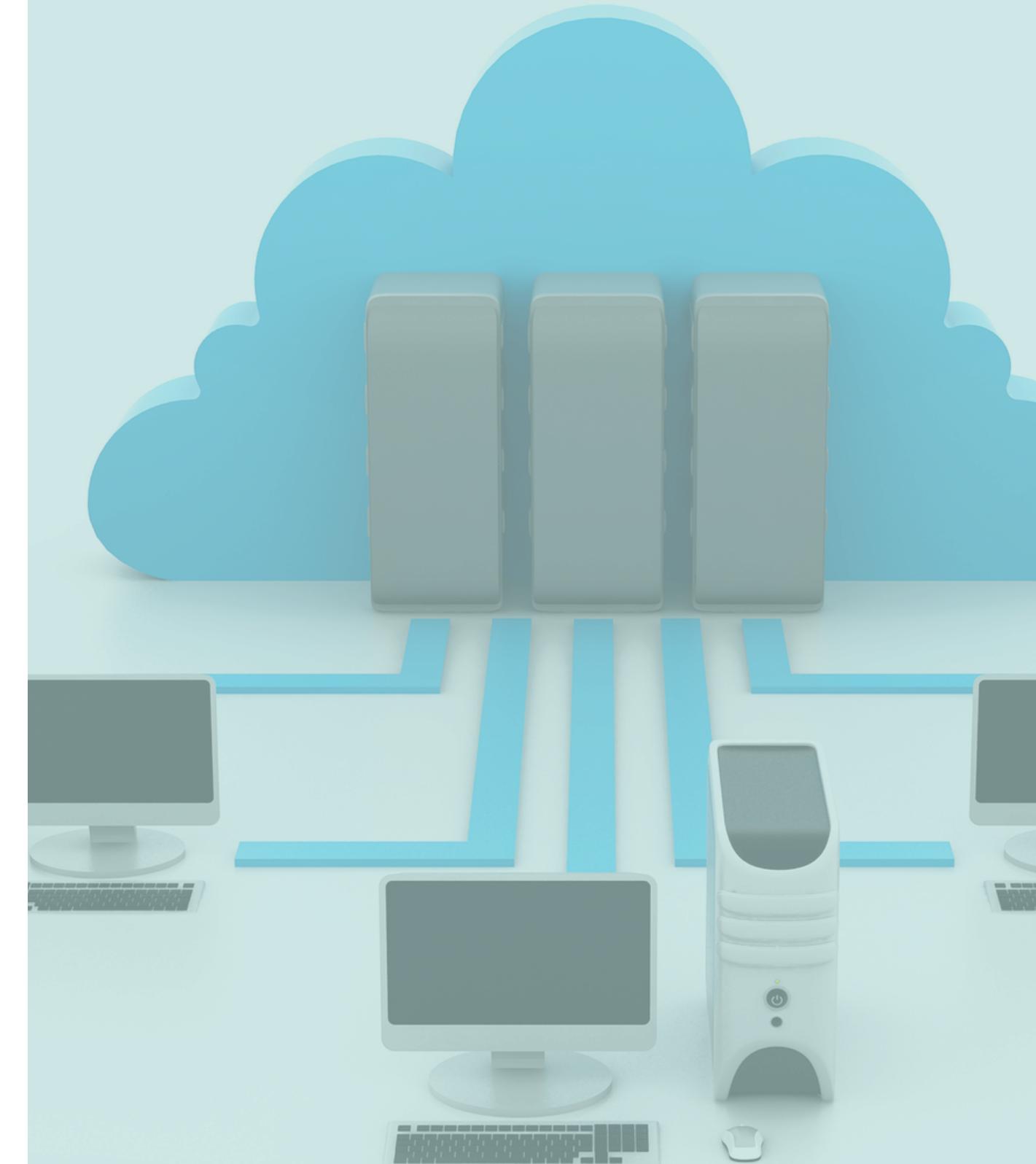
- LXC teve desempenho quase igual ao nativo, mesmo com todos os recursos sendo usados.
- KVM teve desempenho inferior, especialmente em cenários com múltiplas VMs.
- Quanto mais VMs (Virtual-2, Virtual-4), mais o desempenho cai no KVM.



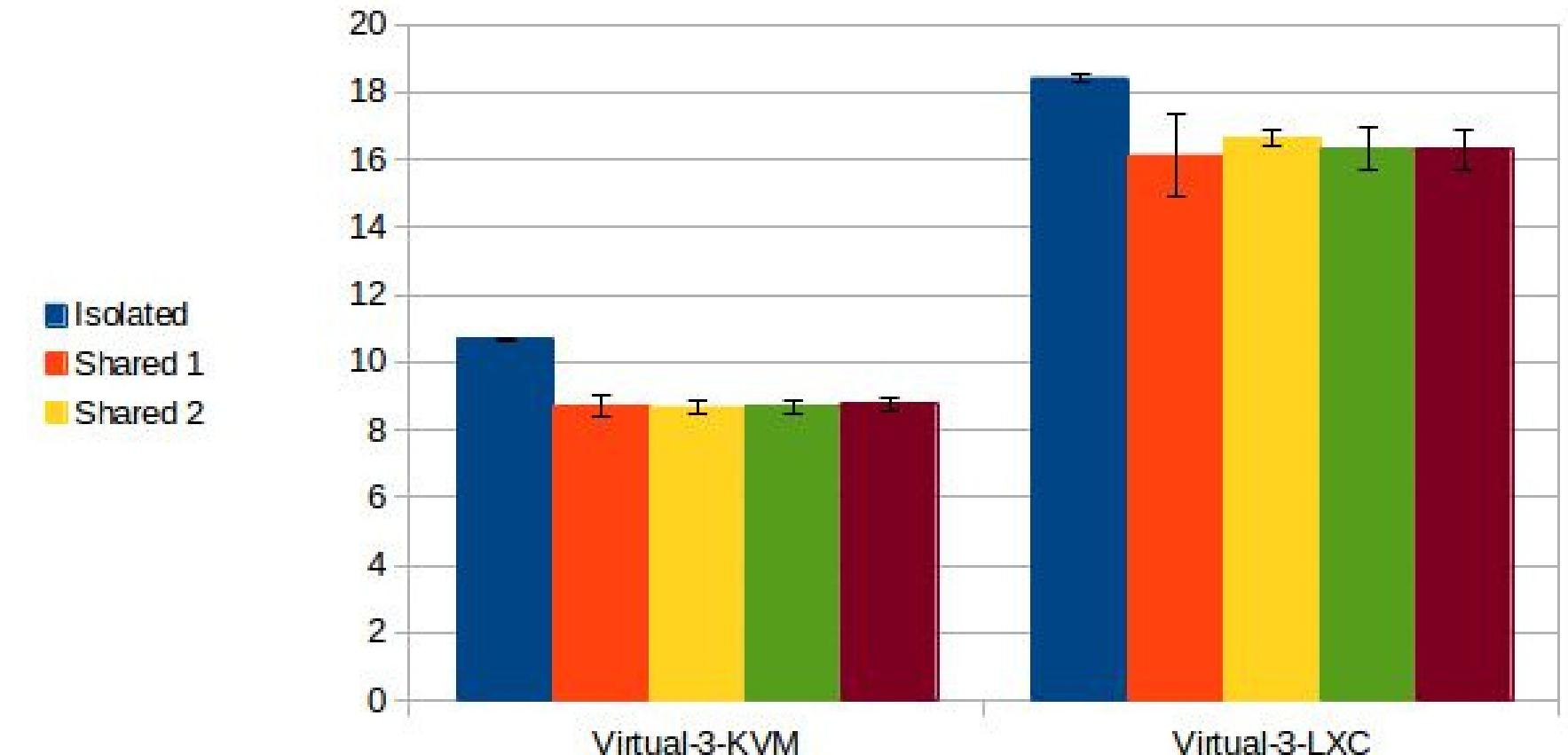
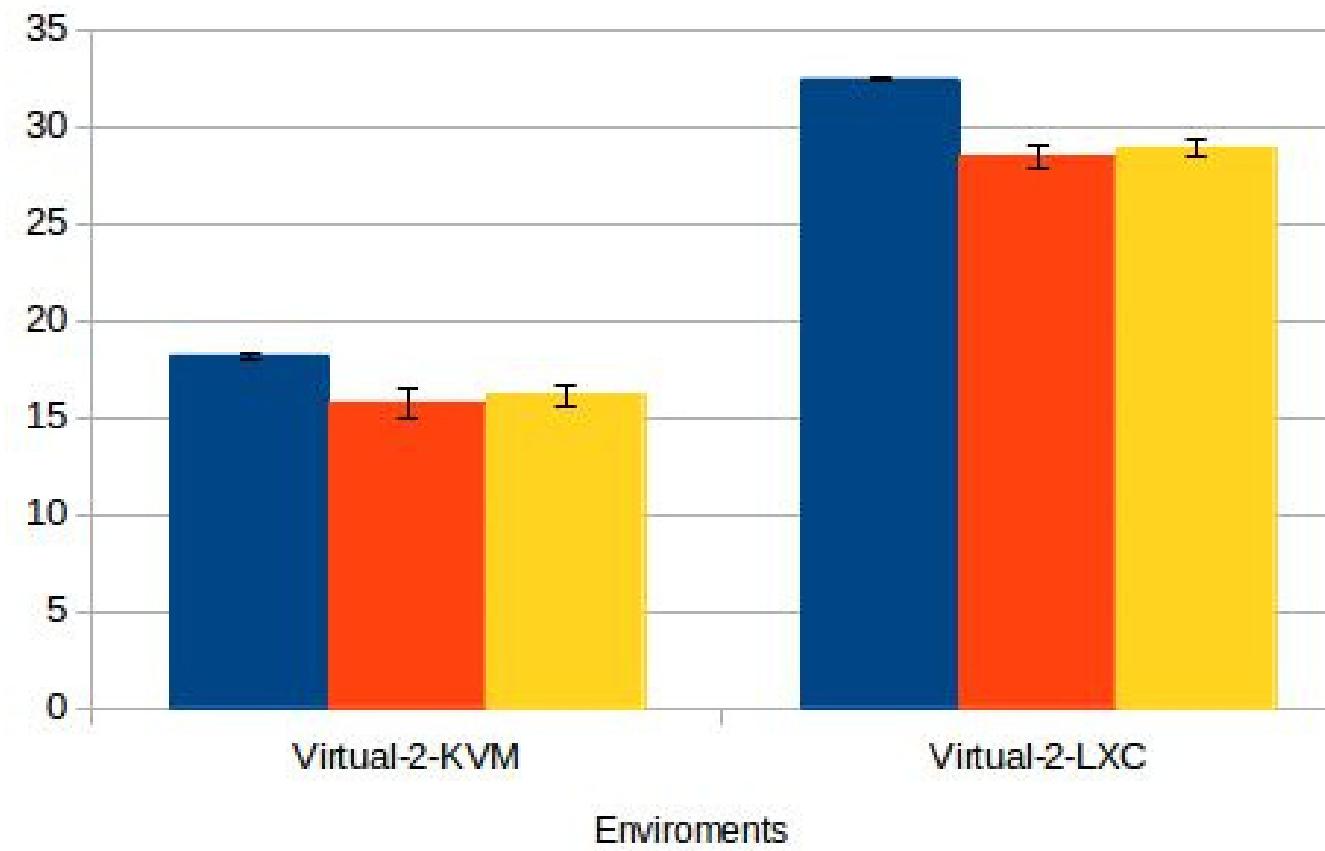
Efeitos do Compartilhamento de Recursos

Benchmark de CPU usando o HPL, com instâncias concorrentes.

- KVM teve queda de desempenho de até 20% com 4 VMs.
- LXC teve queda constante de 11,5% em todas as situações.
- KVM consome mais CPU do host para manter múltiplas VMs.



Efeitos do Compartilhamento de Recursos





53M
13.6M
926.2k
481.2k
221.4k
204k
125.3k
123.4k
78.4k
77.7k

ver todos



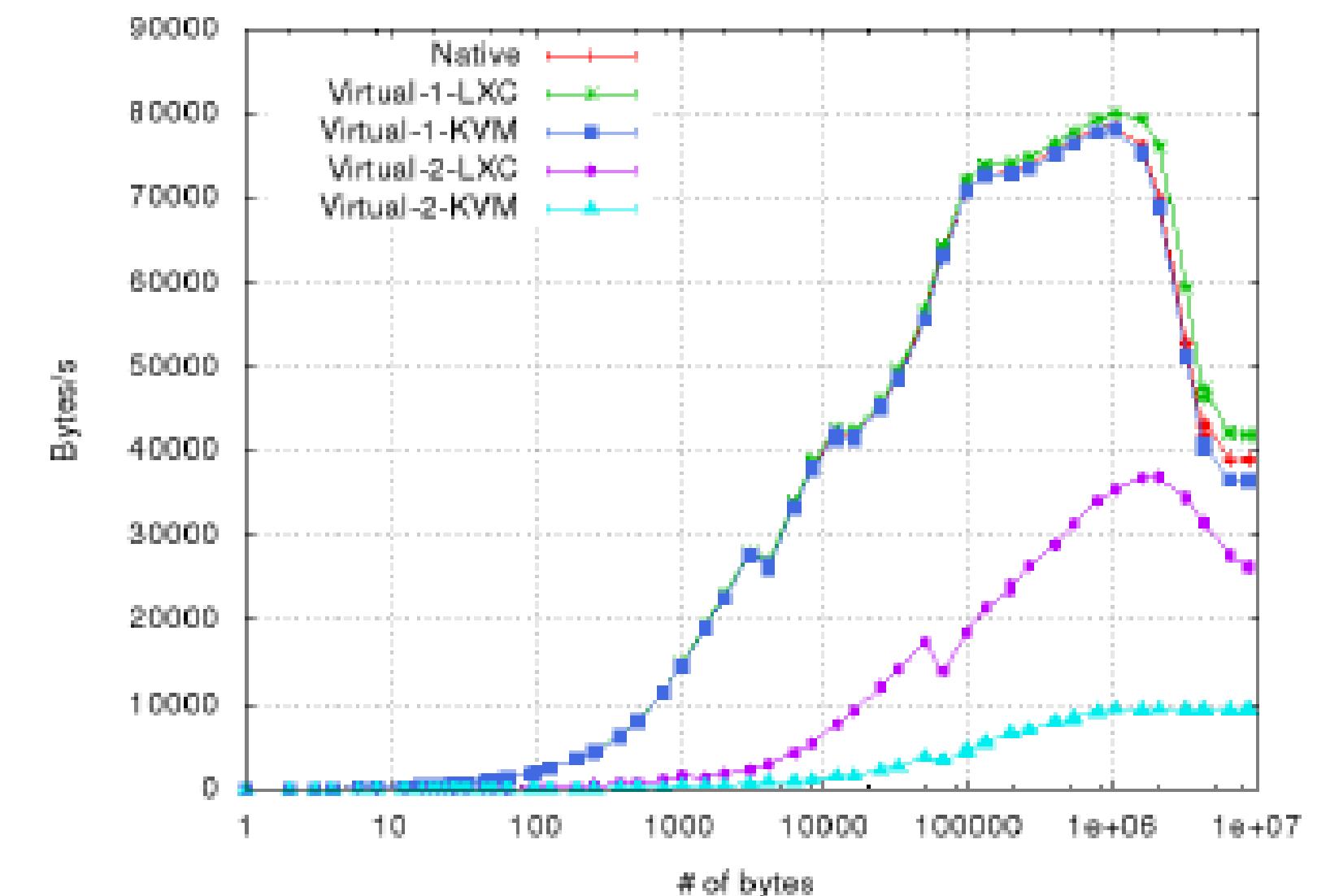
Avaliação de Desempenho de Rede com NetPIPE

- Ferramenta utilizada:
 - **NetPIPE**
- **Objetivo:**
 - Medir largura de banda e latência em diferentes ambientes:
 - Ambiente Nativo
 - Virtualização com LXC e KVM
 - Execução em host único e em cluster

Largura de Banda em Ambiente Local

- Virtual-1 (uma VM): Desempenho similar ao nativo
- Virtual-2 (duas VMs): Redução significativa de banda
- KVM teve maior perda que LXC no Virtual-2

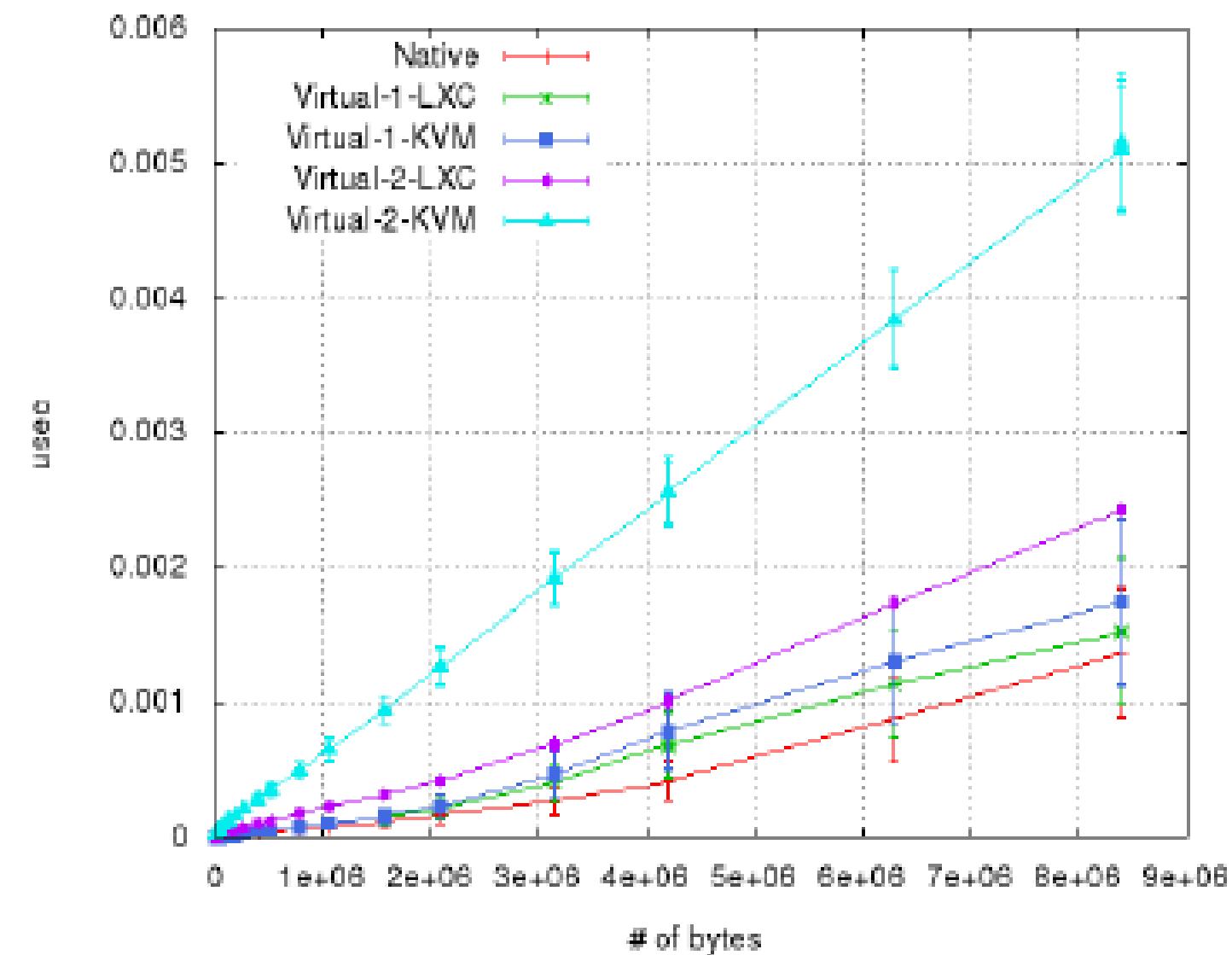
Figure 9 Inter-process communication bandwidth (same host)



Latência de Comunicação em Host Único

- Virtual-2 aumentou significativamente a latência
- Native teve a menor latência
- Variações maiores com KVM

Figure 10 Inter-process communication latency (same host)



Largura de Banda e Latênciam em Cluster

- Pouca variação entre Native, LXC e KVM
- Todos os ambientes apresentaram desempenho semelhante

Figure 11 Inter-process communication bandwidth (cluster)

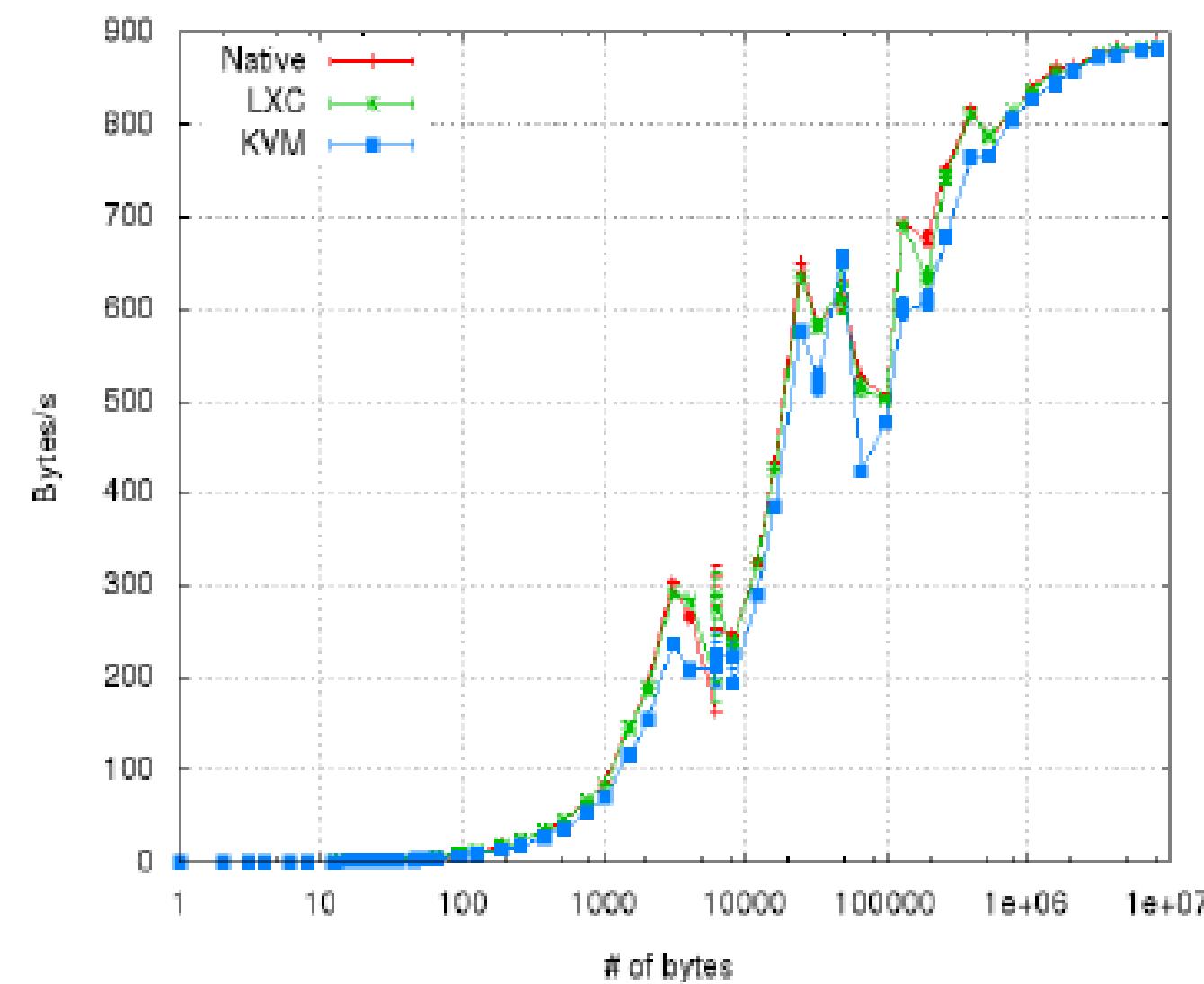
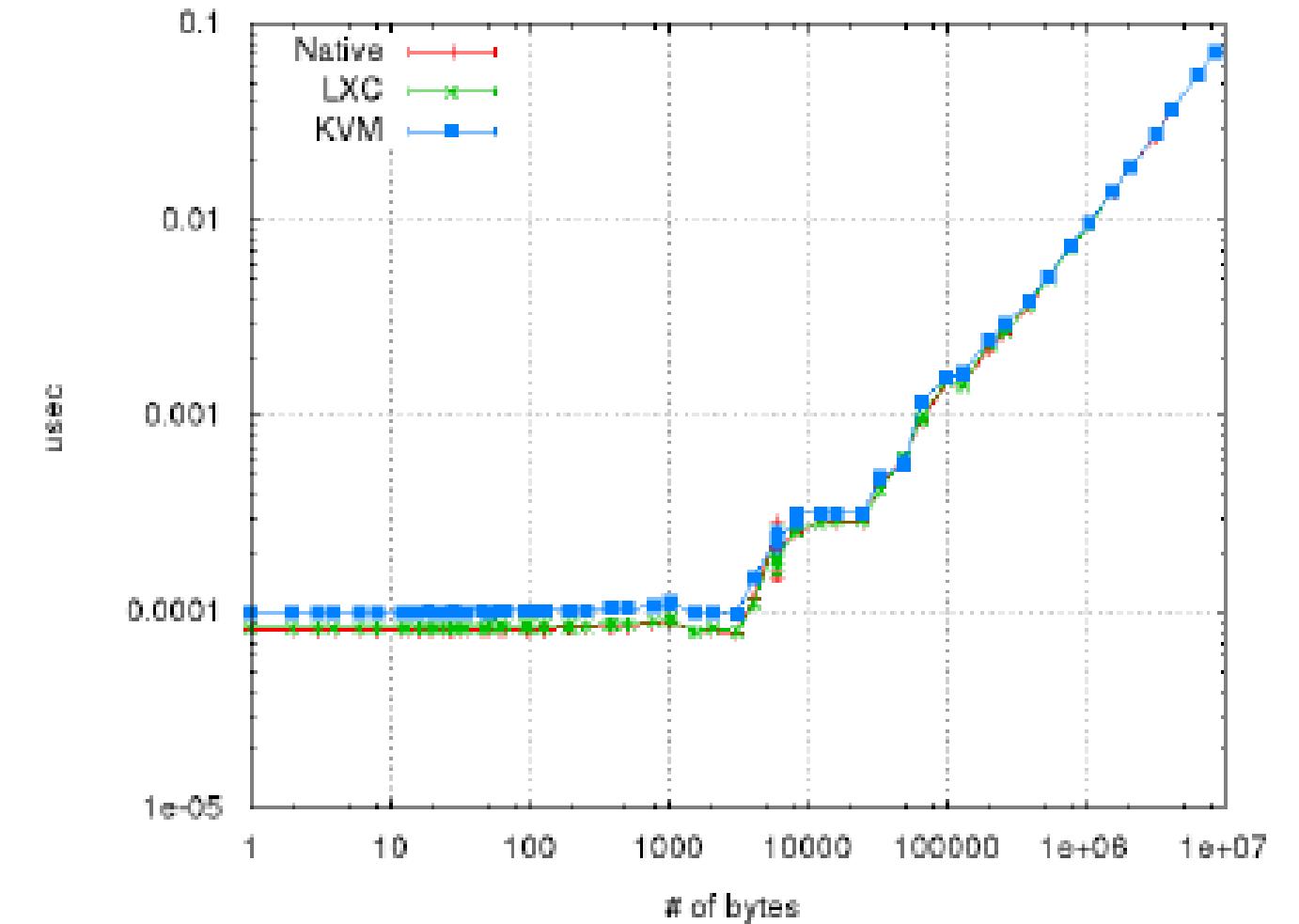


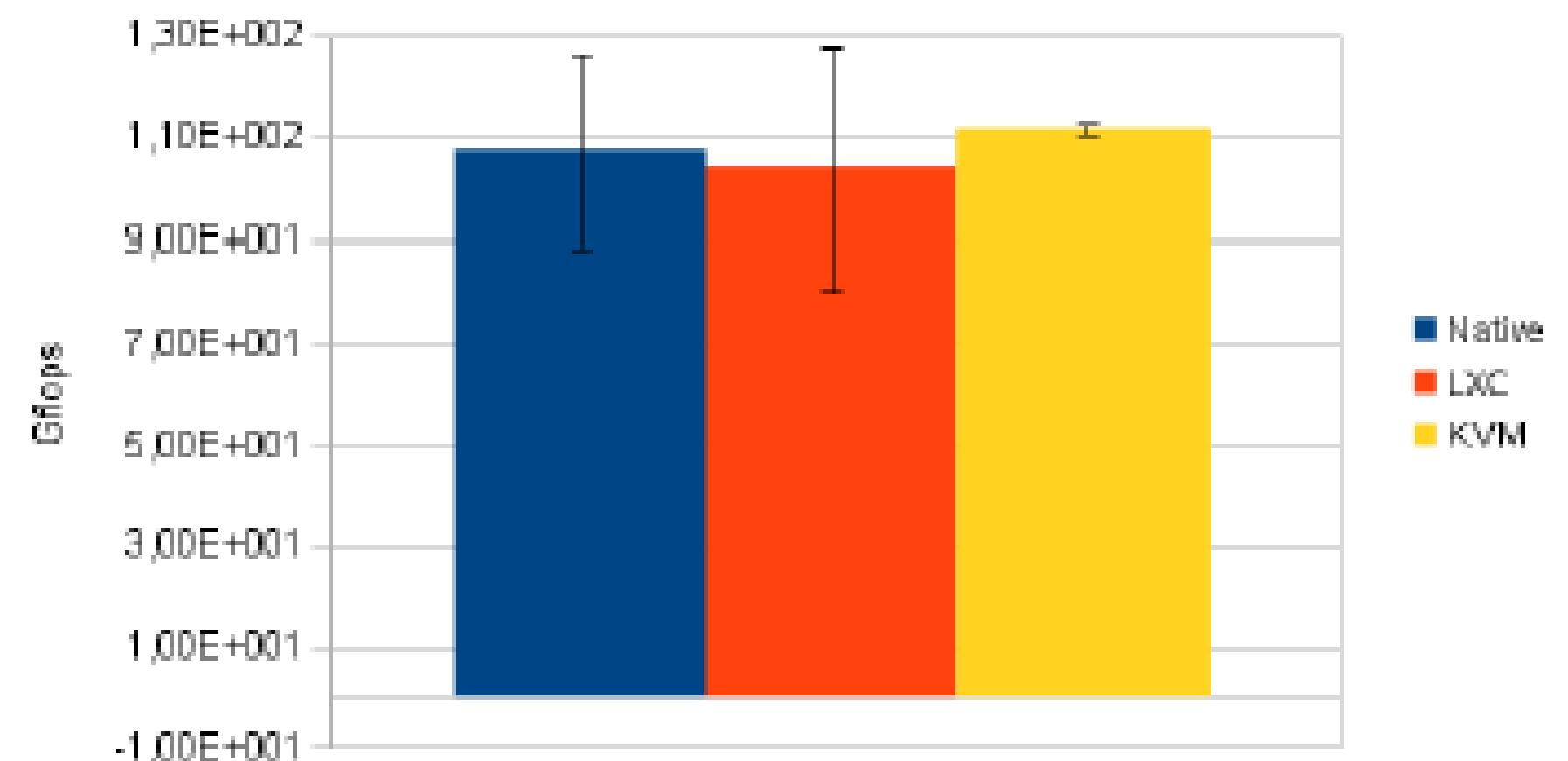
Figure 12 Inter-process communication latency (cluster)

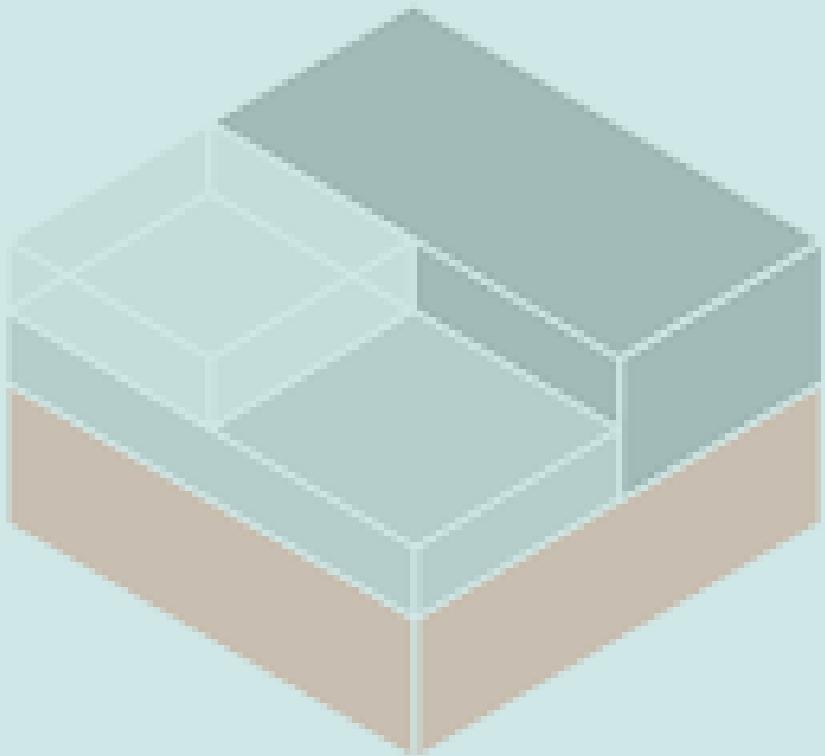


Impacto na Performance HPC(HPL)

- Teste com HPL (High Performance Linpack)
- KVM apresentou menos variações
- Resultados semelhantes entre LXC e KVM

Figure 13 HPL cluster performance (using network)





Problemas encontrados com LXC

- LXC não oferece isolamento completo
- Possível comprometer o host em certas condições-
- Timeout e falha ao atualizar para Jessie

Expectativas e desafios dos contêineres

- Contêineres são promissores (desempenho, leveza)
- Adoção por grandes empresas
- Desafios persistem: segurança, isolamento e compatibilidade





PUC Minas

CONCLUSÃO



Conclusões Principais do Estudo

- Os resultados obtidos nas simulações foram consistentes com o comportamento esperado do sistema.
- A repetição dos testes manteve a estabilidade dos dados, mostrando que o modelo é confiável.
- As decisões de modelagem se mostraram adequadas para representar o sistema de forma realista.

Perspectivas Futuras

- A simulação provou ser uma ferramenta essencial para prever o desempenho de sistemas complexos.
- Parâmetros mal ajustados podem comprometer os resultados, exigindo conhecimento profundo do sistema.
- Extensões futuras podem incluir:
 - Análise de cenários de sobrecarga;
 - Inclusão de falhas no sistema;
 - Avaliação de diferentes políticas de escalonamento.





PUC Minas

MUITO OBRIGADO!

ANDRÉ LUIS SILVA
BRENO SANTOS
CAIO DINIZ
GIUSEPPE CORDEIRO
MATHEUS FAGUNDES
VINÍCIUS MIRANDA
YAN SABARENSE