

**Evolution of TPUs** 

**TPU Evolution** 

| Generation        | Primary Use<br>Case       | Precision                 | Key Features                        |
|-------------------|---------------------------|---------------------------|-------------------------------------|
| TPUv1-v2          | Early Inference           | 8-bit/16-bit              | Efficient matrix computation        |
| TPUv3-v4          | Training and<br>Inference | Higher                    | Greater memory<br>bandwidth         |
| TPUv5/v5p         | Training LLMs             | Mixed (bfloat16,<br>int8) | Optimized for large language models |
| TPUv6e (Trillium) | Large Model<br>Training   | Unknown                   | Cost-efficient<br>training          |
| TPUv7 (Ironwood)  | Inference-First           | Unknown                   | Active reasoning and generation     |
|                   |                           |                           |                                     |

HBM3

Memory

**Ironwood Architecture** 

## Enhancement Integrated Interconnect

Integrated

Interconnect (ICI)

Chip Features

SparseCore

SparseCore

enhances the

performance of

recommender



The Sweet Spot of Al Acceleration

Balanced

Performance

Optimal AI acceleration

Computational

Speed and resource

Efficiency

optimization

FP8 Precision [0]

Accuracy

Precision in Al

computations

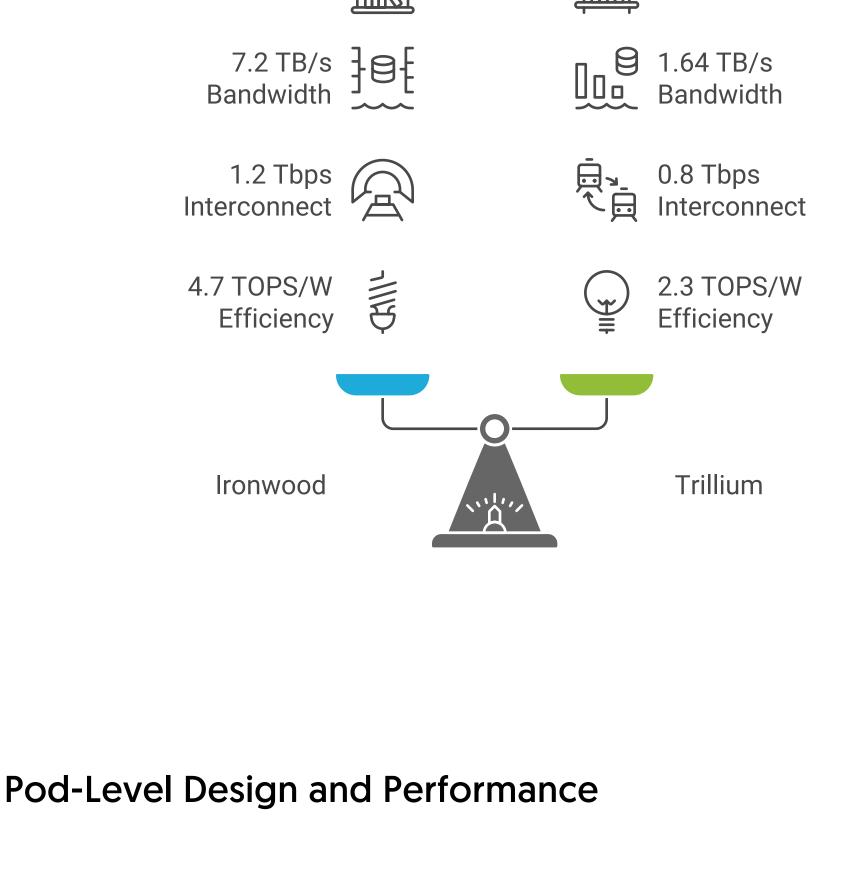
Hardware



**Ironwood outperforms Trillium in key performance** 

metrics.

Bfloat16
Precision



El Capitan

World's fastest

supercomputer with

\~1.7 exaFLOPs.

**Top AI Compute Pod Configurations** 

9,216-Chip Pod

Delivers 42.5

exaFLOPs of AI

compute using FP8.

256-Chip Pod

Common in Google

Cloud deployments.

Torus

efficient

**Topologies** 

communication.

Low Latency

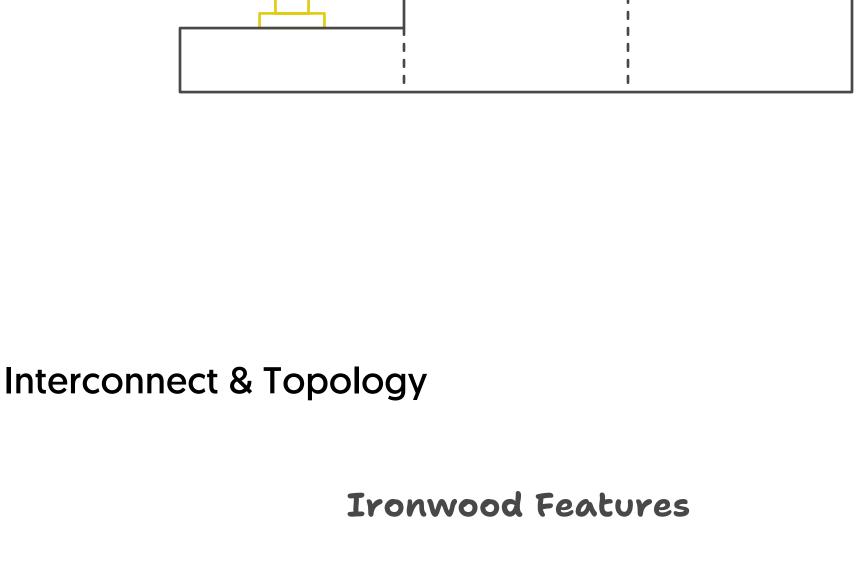
Enables ultra-low

latency token

prediction.

Chips are connected in

torus topologies for



## synchronous largemodel inference. Seamless Scaling

High-Speed

Ironwood uses high-

speed ICI 1.2 Tbps bidirectional mesh

Synchronous

Provides seamless

scaling across pods

with Jupiter optical

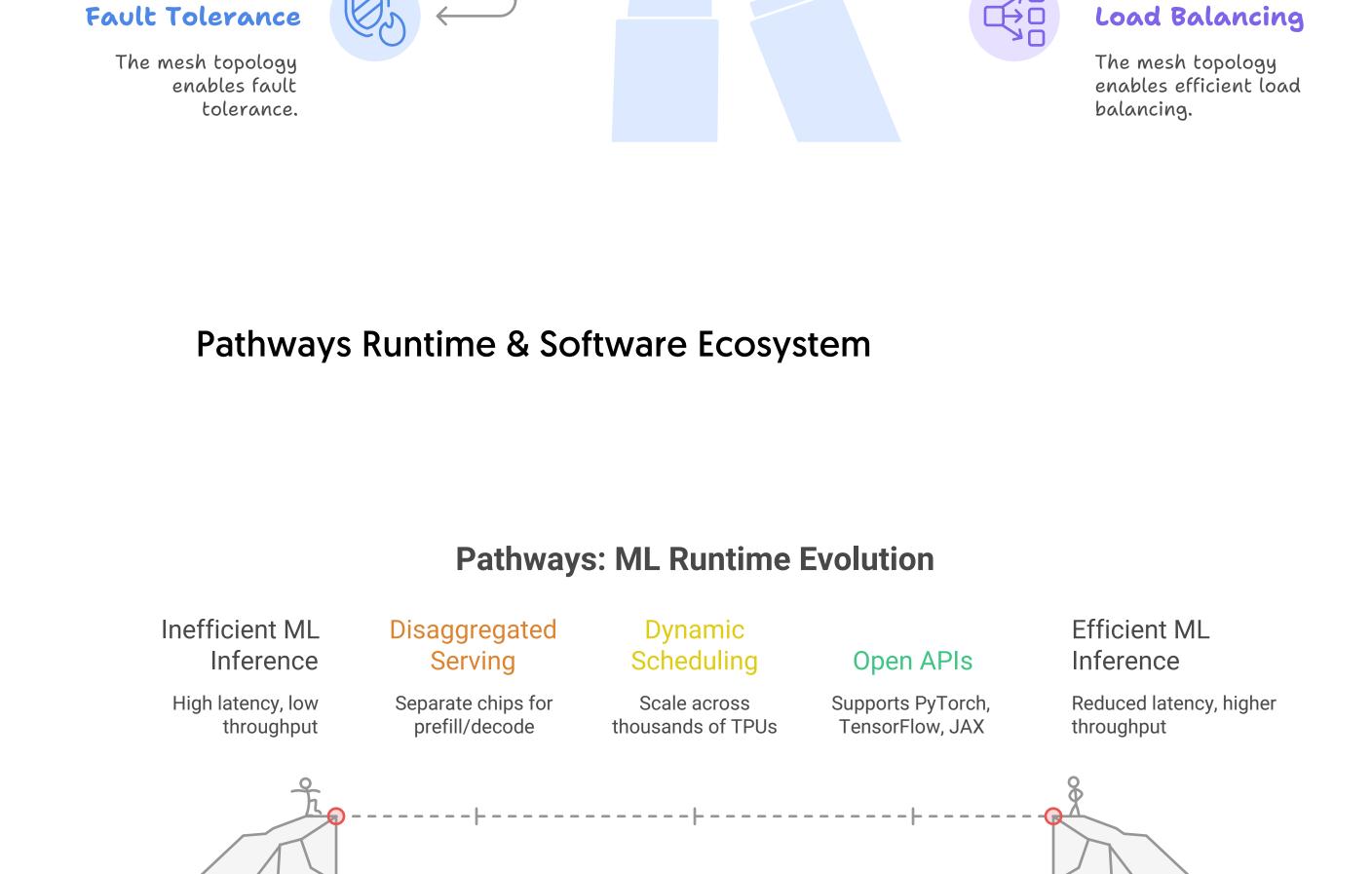
switches.

Inference

Allows for

Links

links.



Which workload should Ironwood be optimized for?

**LLM Inference** 

MoE Models

Optimizing for LLM inference allows for efficient processing of large language

Focusing on MoE models enhances routing and sparse activation, improving efficiency.

Recommender Systems

systems improves performance.

**Scientific Computing** 

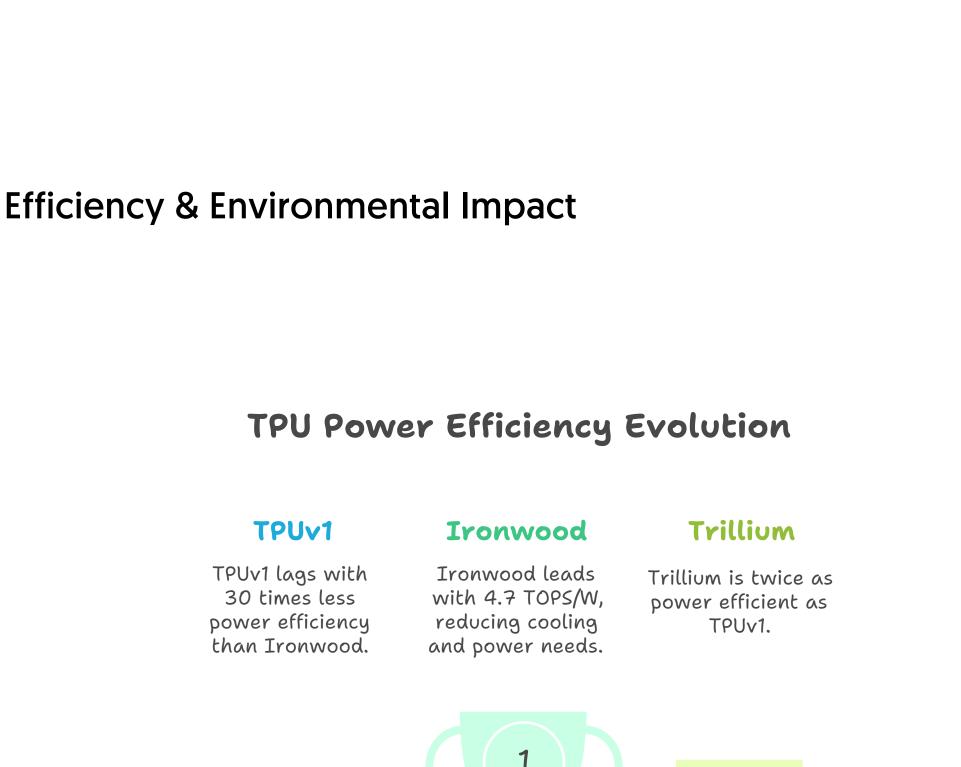
Accelerating embeddings in recommender

Optimizing for scientific computing enables faster processing of financial and DNA

2

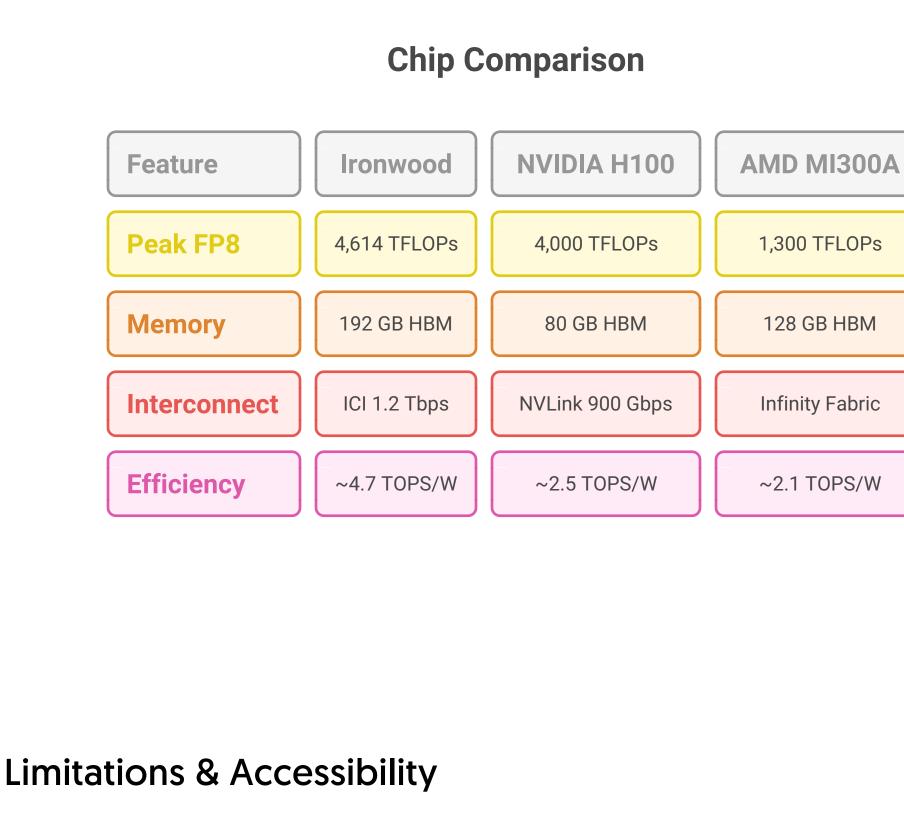
models like PaLM and Gemini.

# Al Use Cases Enabled



**Competitive Comparison** 

3



Limited

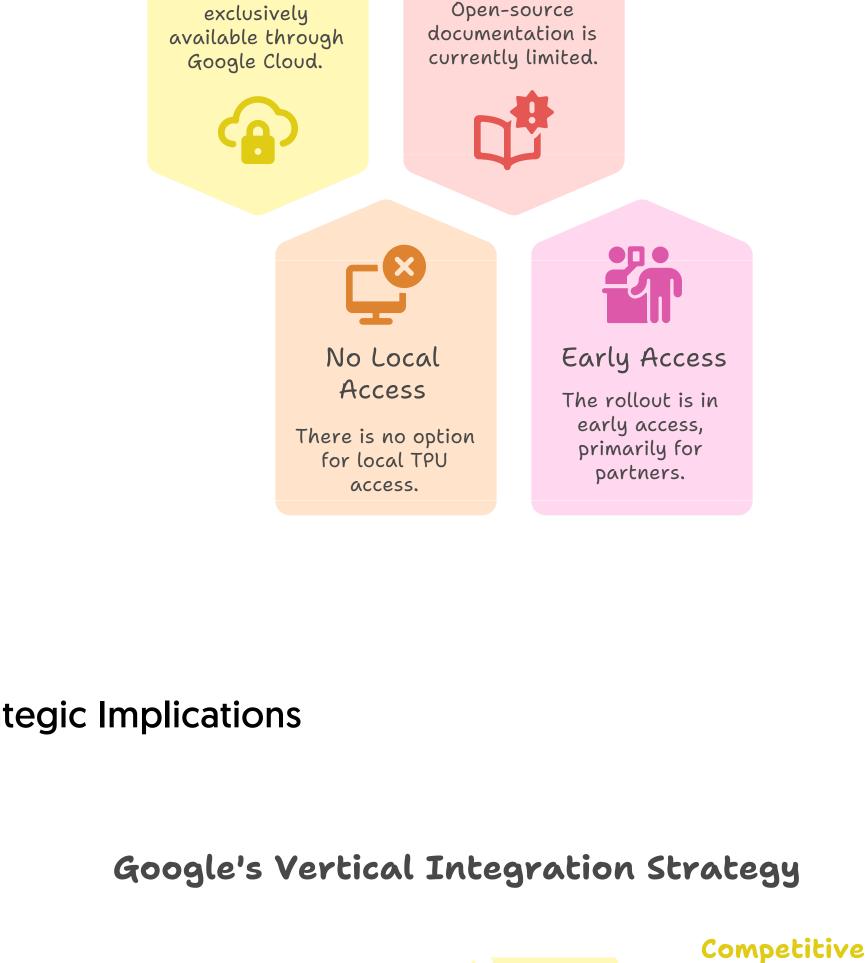
Documentation

Open-source

# **Ironwood Limitations**

Cloud Only

Ironwood is



# Strategic Implications

specialized TPUs

for AI tasks

Cloud Stack stacks Control Managing GCP for seamless Compiler/Runtime integration Developing 3 Pathways for efficient TPU Custom execution Hardware 4 Designing

Edge

Outperforming traditional GPU