GPU推論サーバの待ち行列モデル

井上　文彰

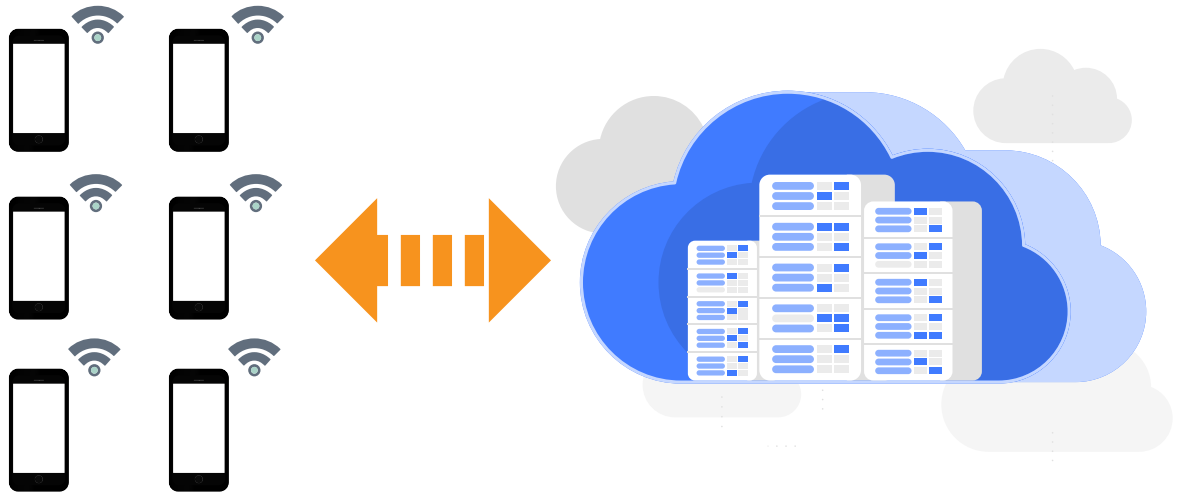
大阪大学大学院工学研究科

機械学習技術

- 近年，機械学習技術 (AI) が目覚ましい発展を遂げている
  - 画像識別，音声認識，自動翻訳，etc.
- 機械学習には 学習 と 推論 の 2 フェーズが存在
  - 学習: 膨大なデータからパターンや構造を抽出
    - 例) 画像識別では，画素値集合からラベル集合への関数 $f : \mathbb{R}^{\mathrm{Length} \times \mathrm{Width}} \to X_{\mathrm{Label}}$ を学習
  - 推論: 学習結果を用いて実際のデータを処理
    - 上記の例では，f を入力画像に作用させることに相当
- AI 機能を使うアプリケーションは，推論フェーズに相当

推論アプリケーション

- 最近では 深層ニューラルネット (DNN) の利用が主流
  - 基本的には，単純なアフィン変換と活性化関数の繰り返し
  - 高精度な推論処理が可能である一方，計算負荷が大きい
- DNN での推論はモバイル端末で実行するには負荷が大きすぎる
  - ➡ クラウドサーバにデータを送り，計算を依頼 (オフロード)



GPU を用いた推論処理

- DNN に基づく推論サーバでは，多くの場合 GPU を利用
  - GPU の並列計算能力を活用
  - 複数のジョブを バッチ化 すると，効率が劇的に向上 [1]
- 文献 [1] で示されている実測結果 (ResNet-50 による画像識別)

Tesla V100 (Mixed precision)

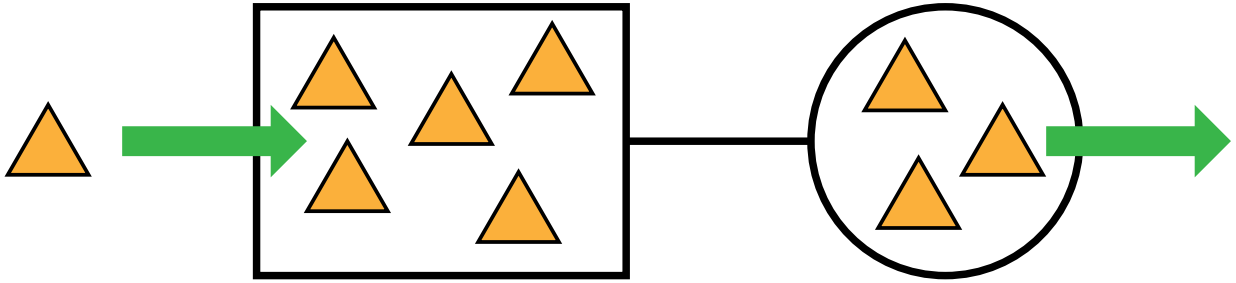
バッチサイズ	スループット [枚/秒]	電力効率 [枚/J]
1	476	4
2	880	8.1
4	1,631	12.4
8	2,685	17.5
64	5,877	21.4
128	6,275	22

Tesla P4 (INT8)

バッチサイズ	スループット [枚/秒]	電力効率 [枚/J]
1	569	12.9
2	736	16.7
4	974	19.9
8	1,291	22.6
64	1,677	26.6
128	1,676	27

[1] Nvidia AI Inference Platform, Giant Leaps in Performance and Efficiency for AI Services, from the Data Center to the Network's Edge.
https://www.nvidia.com/en-us/data-center/resources/inference-technical-overview/

待ち行列モデル化 (1)



- 率 λ のポワソン過程に従ってジョブが到着
- サーバは複数のジョブをバッチ化して処理可能
  - 最大バッチサイズ $M \in \{1, 2, \ldots\} \cup \{\infty\}$
- バッチの処理時間はサイズに依存し，独立に分布
  - $H^{[b]}$: サイズ b のバッチ処理時間を表す確率変数
  - $\mu^{[b]} := \dfrac{b}{\mathrm{E}[H^{[b]}]}$ バッチサイズ b での平均スループット

待ち行列モデル化 (2)

$\mu^{[b]} = \dfrac{b}{\mathrm{E}[H^{[b]}]}$　　バッチサイズ b での平均スループット

- 今回は，次の単純なバッチ化法に限定して考察
  - (M を超えない範囲で) 全ての待機ジョブ をバッチに入れて処理

すなわち，待機ジョブ数 = x のときサイズ $b = \min(x, M)$ で処理開始

- 以降では，$\mu^{[b]}$ に関して次の仮定を置く

(i) b に関して $\mu^{[b]}$ は単調非減少

(ii) $\mu^{[M]} > \lambda$　　($M = \infty$ のとき $\mu^{[M]} := \lim_{b \to \infty} \mu^{[b]}$)

➡ このとき，システムは安定となる

通常の解析法 (1)

$L_{\mathrm{D},n}$: n 番目の 処理完了直後 における待機ジョブ数

B_n: n 番目の処理バッチサイズ

A_n: n 番目のバッチ処理中に到着したジョブ数

- $L_{\mathrm{D},n+1} = L_{\mathrm{D},n} - B_n + \min(1, A_n)$
  - $B_n = \min(L_{\mathrm{D},n}, M)$ バッチサイズは最大限
  - $\Pr(A_n = k) = \int_0^\infty \frac{e^{-\lambda x}(\lambda x)^k}{k!} \mathrm{d}H^{[B_n]}(x)$ 処理時間はサイズ依存

➡ $(L_{\mathrm{D},n})_{n=1,2,\ldots}$ はマルコフ連鎖をなす

通常の解析法 (2)

$L_{\mathrm{D},n}$: n 番目の処理完了直後における待機ジョブ数

- $(L_{\mathrm{D},n})_{n=1,2,\ldots}$ はマルコフ連鎖をなす

最大バッチサイズ $M < \infty$ とすると,

$L_{\mathrm{D},n} \geq M$ に対し, $L_{\mathrm{D},n+1} = L_{\mathrm{D},n} - M + A_n$

$$\Pr(A_n = k) = \int_0^\infty \frac{e^{-\lambda x}(\lambda x)^k}{k!} \mathrm{d}H^{[M]}(x)$$

$L_{\mathrm{D},n}$ に依らない

➡ 状態を M 個ずつグループ化すれば, M/G/1 型マルコフ連鎖

- したがって, アルゴリズム的解法による数値解析が可能

本研究の動機

- アルゴリズム的解法では，性能指標の 陽表現式 は得られない
- 付加的な仮定 を課し，性能指標の陽な特徴付けを行いたい

  (i) サイズ b のバッチ処理時間 $H^{[b]}$ は 一定値 $\tau^{[b]}$ を取る

  (ii) ある $\alpha > 0, \tau_0 \geq 0$ に対し，

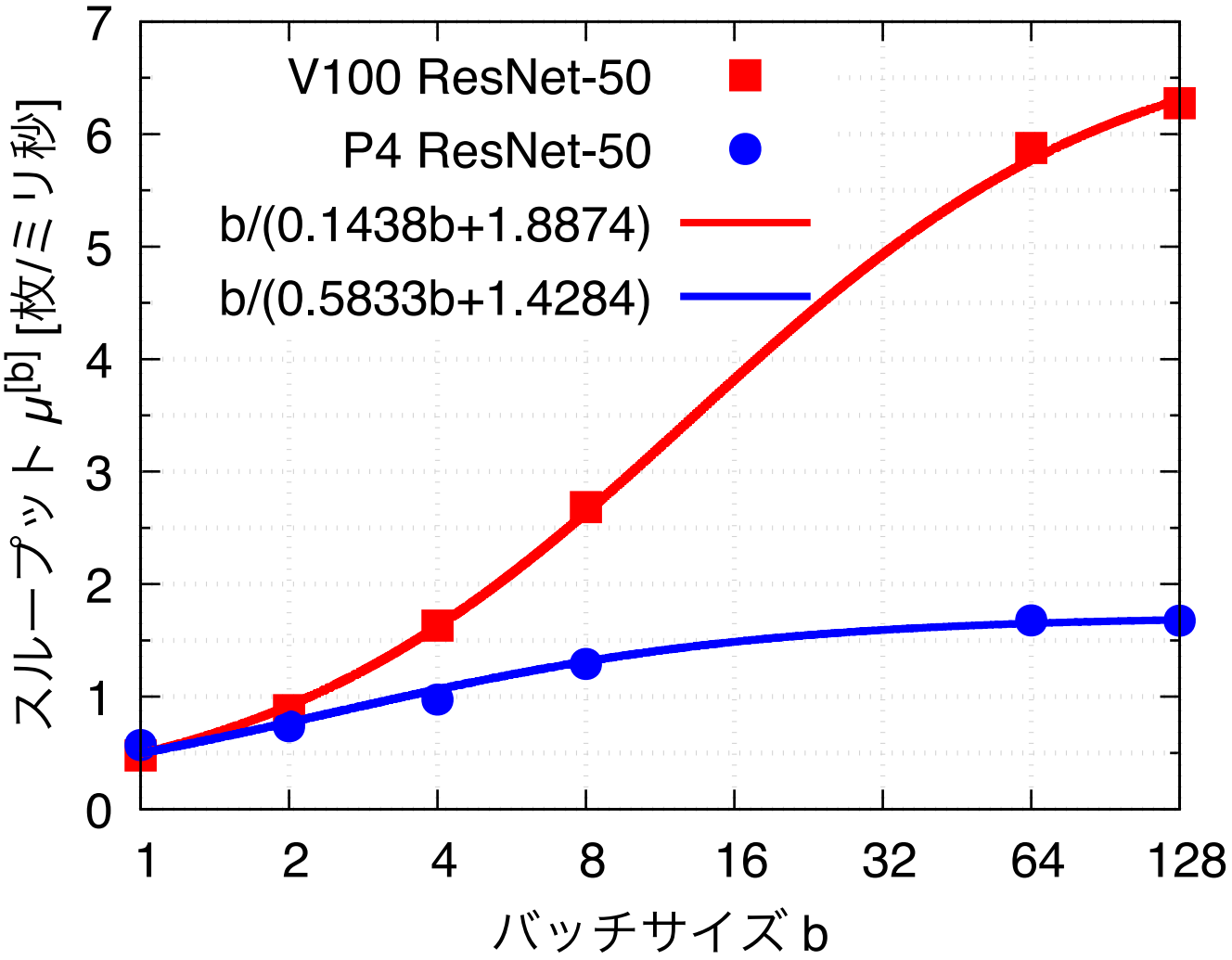
  $$\tau^{[b]} = \alpha b + \tau_0, \quad b = 1, 2, \ldots$$

  (iii) 最大バッチサイズ $M = \infty$

- (i) は DNN を用いた推論では自然な仮定 (計算ステップは入力非依存)
- 冒頭で示したデータは (ii) によく当てはまる (次ページ)
- M は通常，非常に大きい値 (≥ 1000) を取る

GPU による推論スループット

前掲のデータについて，スループット $\mu^{[b]}$ を計算しプロット



- 近似曲線

$$\mu^{[b]} = \frac{b}{\alpha b + \tau_0}$$

に良く当てはまる

- α および τ_0 は，バッチ処理時間

$$\tau^{[b]} = \alpha b + \tau_0$$

の最小二乗法で決定

V100: $R^2 \simeq 0.99978$

P4: $R^2 \simeq 0.99998$

本発表の概要

- GPU 推論サーバを表す待ち行列モデル
  - ジョブは率 λ のポワソン到着
  - サイズ b のバッチ処理時間 $H^{[b]}$ は一定値 $\tau^{[b]}$ を取る
    - $\tau^{[b]} = \alpha b + \tau_0, \quad b = 1, 2, \ldots$
  - サービス開始時点で全ての待機中ジョブをバッチ化 $(M = \infty)$
- $\mu^{[b]}$: バッチサイズ b でのスループット

$$\mu^{[b]} = \frac{b}{\alpha b + \tau_0} \to \frac{1}{\alpha} \quad (b \to \infty)$$

安定条件 $\rho := \lambda\alpha < 1$

- このモデルに対し，平均遅延時間の単純な上界値公式を導出
- 数値実験により，この上界値が真値の良い近似であることを示す

平均遅延時間の解析

系内ジョブ数分布

L_t: 時刻 t における系内ジョブ数

$\hat{L}_{\mathrm{A},n}$: n 番目のジョブの 到着直前 における系内ジョブ数

$\hat{L}_{\mathrm{D},n}$: n 番目のジョブの 離脱直後 における系内ジョブ数

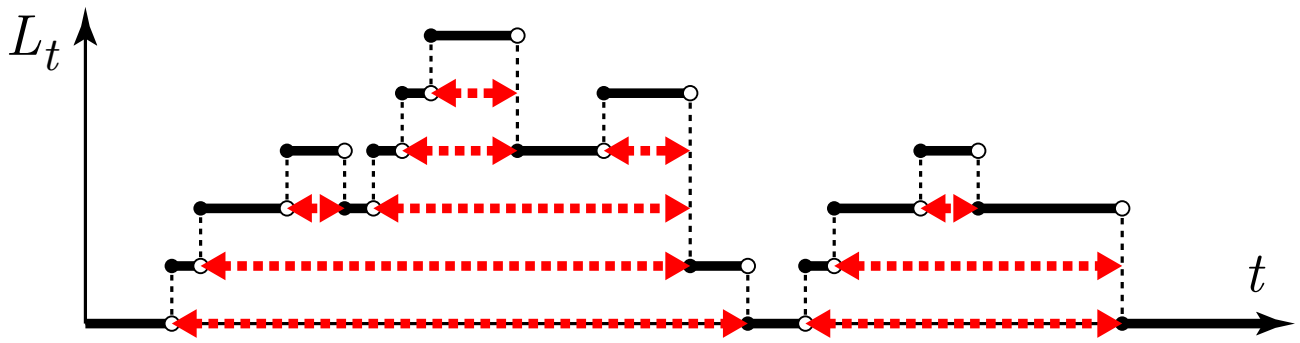
$+$ $(n+1)$ 番目以後に到着 & 同じバッチで処理されるジョブ数

同時に離脱するジョブもそれぞれ区別して扱う

- 定常状態における上記変数を L, $\hat{L}_{\mathrm{A}}$, $\hat{L}_{\mathrm{D}}$ と表すと，次式が成立

$$L =_{\mathrm{st}} \hat{L}_{\mathrm{A}} =_{\mathrm{st}} \hat{L}_{\mathrm{D}}$$

PASTA および下図の観察より



平均遅延時間

L: 定常状態における系内ジョブ数

B: 定常状態における処理バッチサイズ

- $L =_{\mathrm{st}} \hat{L}_{\mathrm{D}}$ を利用し，ジョブの離脱時点に注目すると

$$\mathrm{E}[L] = \sum_{b=1}^{\infty} \frac{b \Pr(B=b)}{\mathrm{E}[B]} \cdot \left(\frac{b-1}{2} + \lambda \mathrm{E}[H^{[b]}] \right)$$

- $\frac{b \Pr(B=b)}{\mathrm{E}[B]}$: ジョブが属するバッチのサイズ
- $\frac{b-1}{2}$: 自分より後に到着したジョブ
- $\lambda \mathrm{E}[H^{[b]}]$: バッチの処理中に到着したジョブ

平均遅延時間

L: 定常状態における系内ジョブ数

B: 定常状態における処理バッチサイズ

- $L =_{\mathrm{st}} \hat{L}_{\mathrm{D}}$ を利用し，ジョブの離脱時点に注目すると

$$\mathrm{E}[L] = \sum_{b=1}^{\infty} \frac{b\Pr(B=b)}{\mathrm{E}[B]} \cdot \left(\frac{b-1}{2} + \lambda \mathrm{E}[H^{[b]}] \right)$$

- リトルの公式より，平均遅延時間 (滞在時間) $\mathrm{E}[W]$ は

$$\mathrm{E}[W] = \frac{\mathrm{E}[B^2] - \mathrm{E}[B]}{2\lambda \mathrm{E}[B]} + \mathrm{E}[\hat{H}]$$

ただし，$\mathrm{E}[\hat{H}] := \sum_{b=1}^{\infty} \frac{b\Pr(B=b)}{\mathrm{E}[B]} \cdot \mathrm{E}[H^{[b]}]$

処理時間分布に関する仮定

$$\mathrm{E}[W] = \frac{\mathrm{E}[B^2] - \mathrm{E}[B]}{2\lambda \mathrm{E}[B]} + \mathrm{E}[\hat{H}]$$

- ここで，バッチ処理時間 $H^{[b]}$ $(b = 1, 2, \ldots)$ に関する前述の仮定

$$H^{[b]} = \alpha b + \tau_0, \quad \text{w.p. 1}$$

を用いると次式を得る

$$\mathrm{E}[\hat{H}] = \sum_{b=1}^{\infty} \frac{b \Pr(B = b)}{\mathrm{E}[B]} \cdot (\alpha b + \tau_0) = \frac{\alpha \mathrm{E}[B^2]}{\mathrm{E}[B]} + \tau_0$$

➡ 平均遅延時間は $\mathrm{E}[B]$ と $\mathrm{E}[B^2]$ を用いて表される

$$\mathrm{E}[W] = \alpha + \tau_0 + \frac{(1 + 2\lambda\alpha)(\mathrm{E}[B^2] - \mathrm{E}[B])}{2\lambda \mathrm{E}[B]}$$

バッチサイズがなすマルコフ連鎖

B_n: n 番目の処理バッチサイズ

A_n: n 番目のバッチ処理中に到着したジョブ数

- $B_{n+1} = \min(1, A_n)$　　全ての待機ジョブをバッチ化
  - $\Pr(A_n = k) = \dfrac{e^{-\lambda(\alpha B_n + \tau_0)}(\lambda(\alpha B_n + \tau_0))^k}{k!} =: a_k^{[B_n]}$

➡ $(B_n)_{n=1,2,\ldots}$ は (G/G/1 型) マルコフ連鎖をなす

遷移確率行列　$\boldsymbol{P} = \begin{pmatrix} a_0^{[1]} + a_1^{[1]} & a_2^{[1]} & a_3^{[1]} & \cdots \\ a_0^{[2]} + a_1^{[2]} & a_2^{[2]} & a_3^{[2]} & \cdots \\ a_0^{[3]} + a_1^{[3]} & a_2^{[3]} & a_3^{[3]} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$

バッチサイズと到着ジョブ数

B_n: n 番目の処理バッチサイズ

A_n: n 番目のバッチ処理中に到着したジョブ数

- $B_{n+1} = A_n + \mathbb{1}\{A_n = 0\}$
  - $\Pr(A_n = k) = \dfrac{e^{-\lambda(\alpha B_n + \tau_0)}(\lambda(\alpha B_n + \tau_0))^k}{k!}$
- 定常状態を考えると,

$$\begin{aligned}\mathrm{E}[B] &= \mathrm{E}[A] + \Pr(A=0) \\ &= \lambda(\alpha \mathrm{E}[B] + \tau_0) + \Pr(A=0)\end{aligned}$$

$$\begin{aligned}\mathrm{E}[B^2] &= \mathrm{E}[A^2] + \Pr(A=0) \\ &= \lambda(\alpha \mathrm{E}[B] + \tau_0) \\ &\quad + \mathrm{E}[\lambda^2(\alpha B + \tau_0)^2] + \Pr(A=0)\end{aligned}$$

➡ $$\mathrm{E}[B] = \frac{\lambda\tau_0 + \Pr(A=0)}{1-\lambda\alpha}, \qquad \mathrm{E}[B^2] = \frac{(1+2\lambda^2\alpha\tau_0)\mathrm{E}[B] + \lambda^2\tau_0^2}{1-\lambda^2\alpha^2}$$

$\mathrm{E}[B]$ と稼働率の関係

B: 定常状態における処理バッチサイズ

- π_0: 定常状態において 系が空である確率 ($1-\pi_0$ はサーバ稼働率)
- リトルの公式を「処理中バッチ数」に適用

$$1-\pi_0 = \frac{\lambda}{\mathrm{E}[B]} \cdot (\alpha \mathrm{E}[B] + \tau_0)$$

- これを用いて前述の式を書き換えると

$$\mathrm{E}[B] = \frac{\lambda\tau_0}{1-\pi_0-\lambda\alpha}, \qquad \mathrm{E}[B^2] = \frac{(1+2\lambda^2\alpha\tau_0)\mathrm{E}[B] + \lambda^2\tau_0^2}{1-\lambda^2\alpha^2}$$

平均遅延時間 $\mathrm{E}[W]$

- 平均遅延時間は $\mathrm{E}[B]$ と $\mathrm{E}[B^2]$ を用いて表される

$$\mathrm{E}[W] = \alpha + \tau_0 + \frac{(1+2\lambda\alpha)(\mathrm{E}[B^2]-\mathrm{E}[B])}{2\lambda\mathrm{E}[B]}$$

- これに前述の式を代入すると，次式を得る

$$\mathrm{E}[W] = \alpha + \tau_0 + \frac{\lambda(1+2\lambda\alpha)\left(2\alpha\tau_0+\alpha^2+\dfrac{(1-\pi_0-\lambda\alpha)\tau_0}{\lambda}\right)}{2(1-\lambda^2\alpha^2)}$$

➡ π_0 を下界値に置き換えることで，$\mathrm{E}[W]$ の上界値が得られる

平均遅延時間の上界

$$\mathrm{E}[W] = \alpha + \tau_0 + \frac{\lambda(1+2\lambda\alpha)\left(2\alpha\tau_0 + \alpha^2 + \dfrac{(1-\pi_0-\lambda\alpha)\tau_0}{\lambda}\right)}{2(1-\lambda^2\alpha^2)}$$

- π_0 を下界値に置き換えることで，$\mathrm{E}[W]$ の上界値が得られる

  (i) $\mathrm{E}[B] = \dfrac{\lambda\tau_0}{1-\pi_0-\lambda\alpha}$ および $\mathrm{E}[B] \geq 1$ より，$\pi_0 \geq 1-\lambda(\alpha+\tau_0)$

  (ii) 自明な下界値　$\pi_0 \geq 0$

これらをそれぞれ用いると，

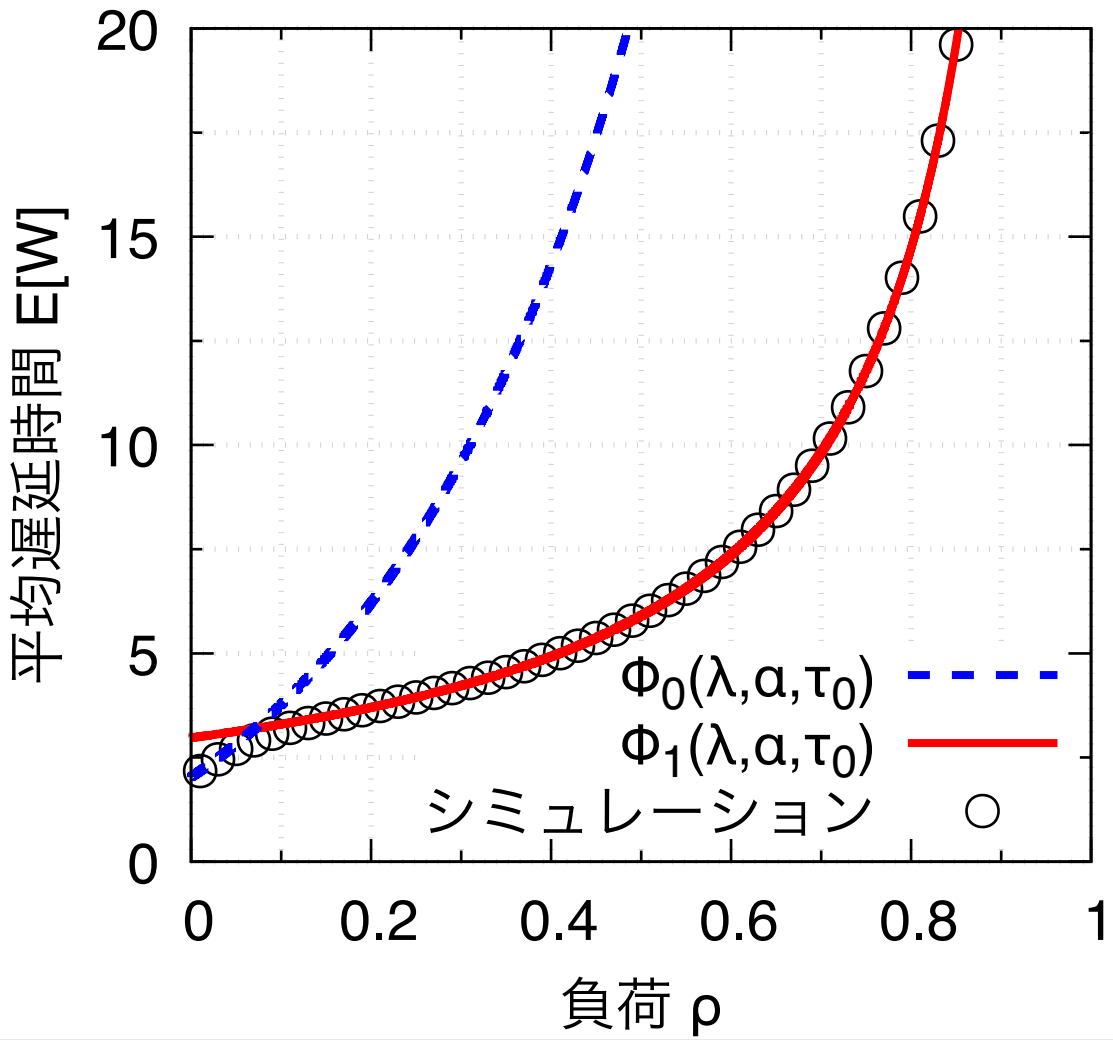
(i) $\mathrm{E}[W] \leq \dfrac{\alpha+\tau_0}{2(1-\lambda\alpha)}\left(1+2\lambda\tau_0+\dfrac{1-\lambda\tau_0}{1+\lambda\alpha}\right) =: \phi_0(\lambda,\alpha,\tau_0)$

(ii) $\mathrm{E}[W] \leq \dfrac{3}{2}\cdot\dfrac{\tau_0}{1-\lambda\alpha} + \dfrac{\alpha}{2}\cdot\dfrac{\lambda\alpha+2}{1-\lambda^2\alpha^2} =: \phi_1(\lambda,\alpha,\tau_0)$

数値評価

平均遅延時間の真値と上界値 (Tesla V100)

Tesla V100 (Mixed Precision)

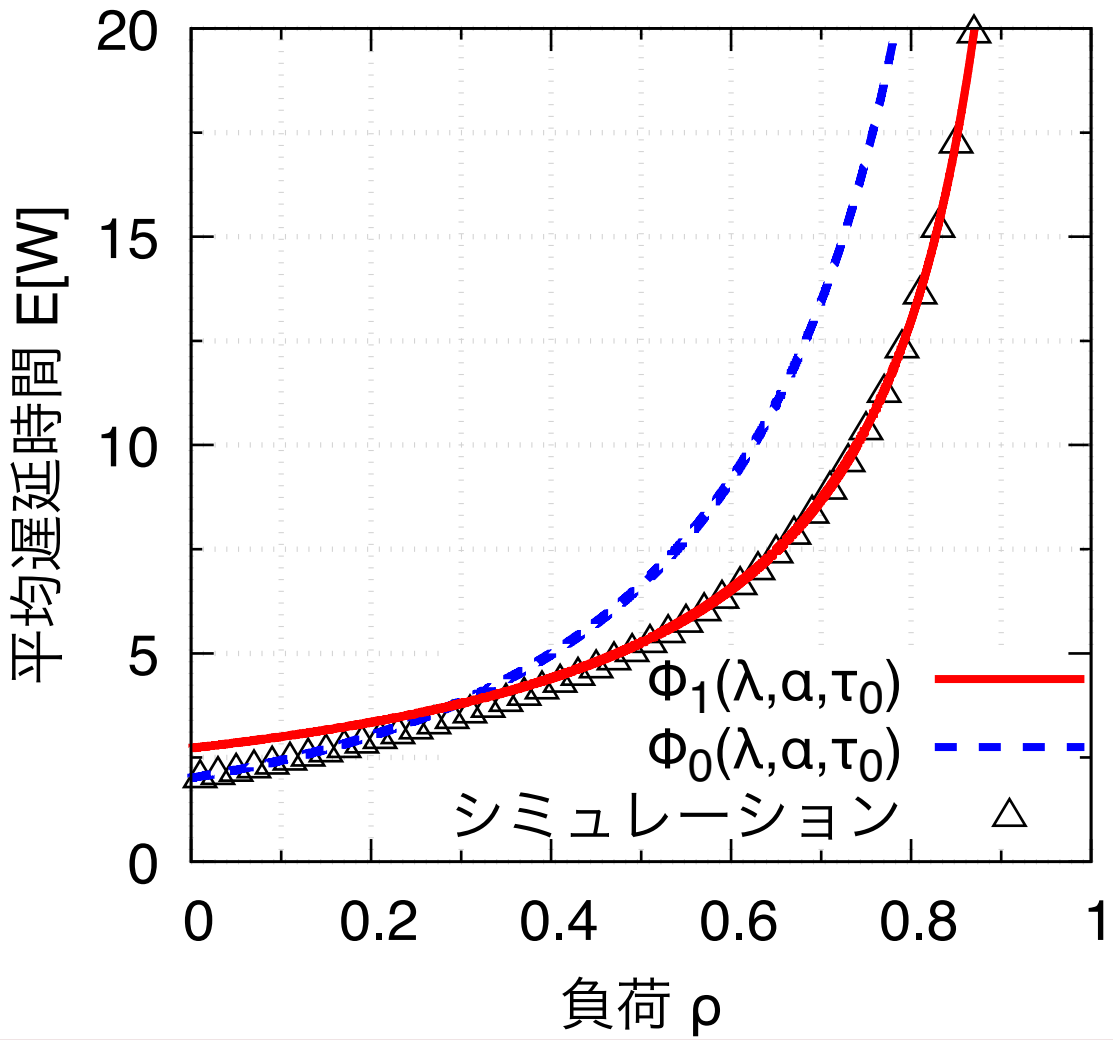


負荷 $\rho := \lambda / \mu^{[\infty]}$

- $\mathrm{E}[W]$ の上界値は
  真値の非常に良い近似
- ρ が小さいとき
  - ϕ_0 が良い近似
  - $\mathrm{E}[B] \simeq 1$ に相当
- それ以外の範囲
  - ϕ_1 が良い近似
  - $\pi_0 \simeq 0$ に相当

平均遅延時間の真値と上界値 (Tesla P4)

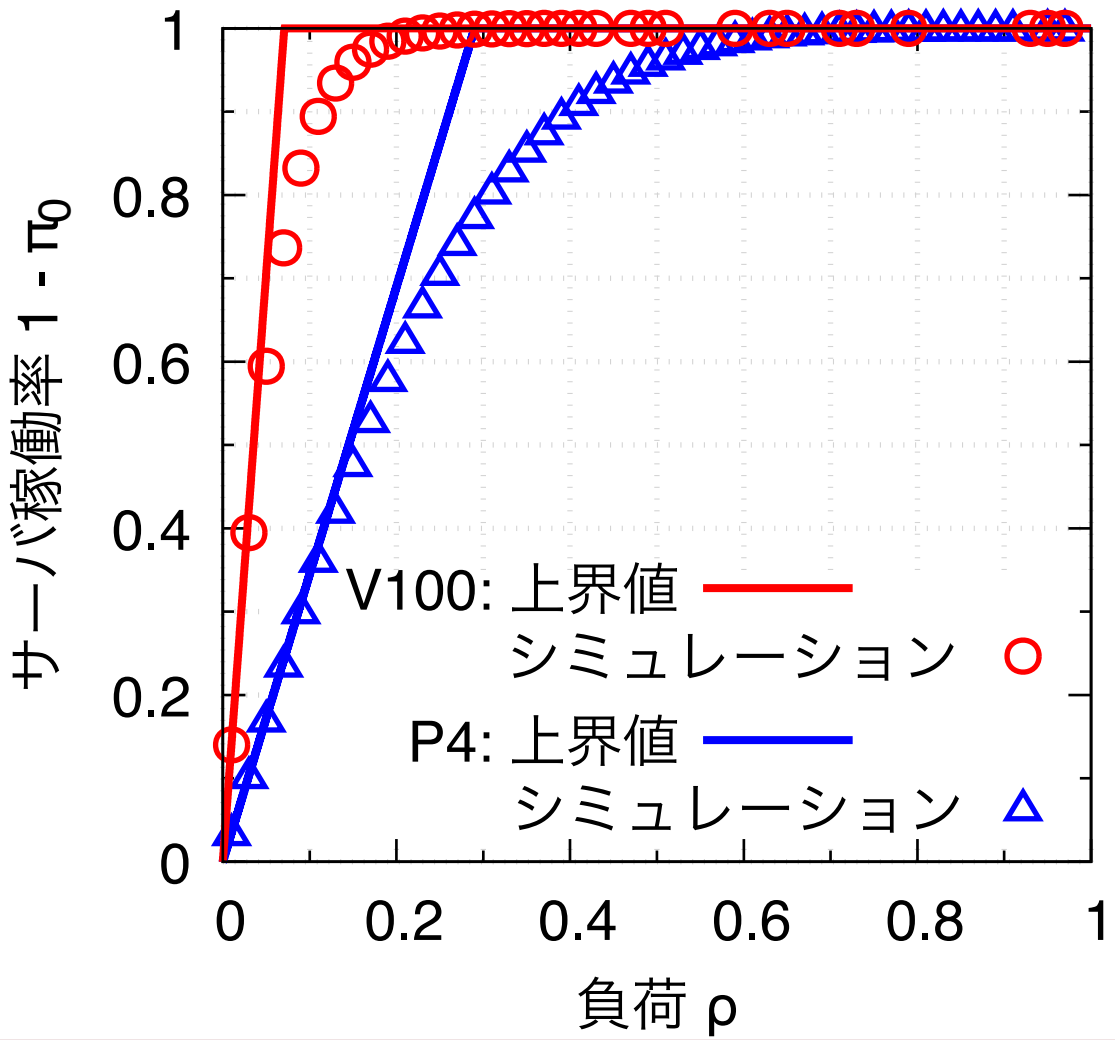
Tesla P4 (INT8)



- Tesla V100 (前ページ) の結果とほぼ同様
- $\rho = 0.3$ の付近では誤差がやや大きい
  - $\mathrm{E}[B] \simeq 1$ の領域と $\pi_0 \simeq 0$ の領域との狭間に相当

サーバ稼働率の真値と上界値

サーバ稼働率 $1-\pi_0 \le \min\left(1, \lambda(\alpha+\tau_0)\right)$



- ρ の値が中程度でも稼働率は 1 に近い
- バッチサイズ b でのスループット

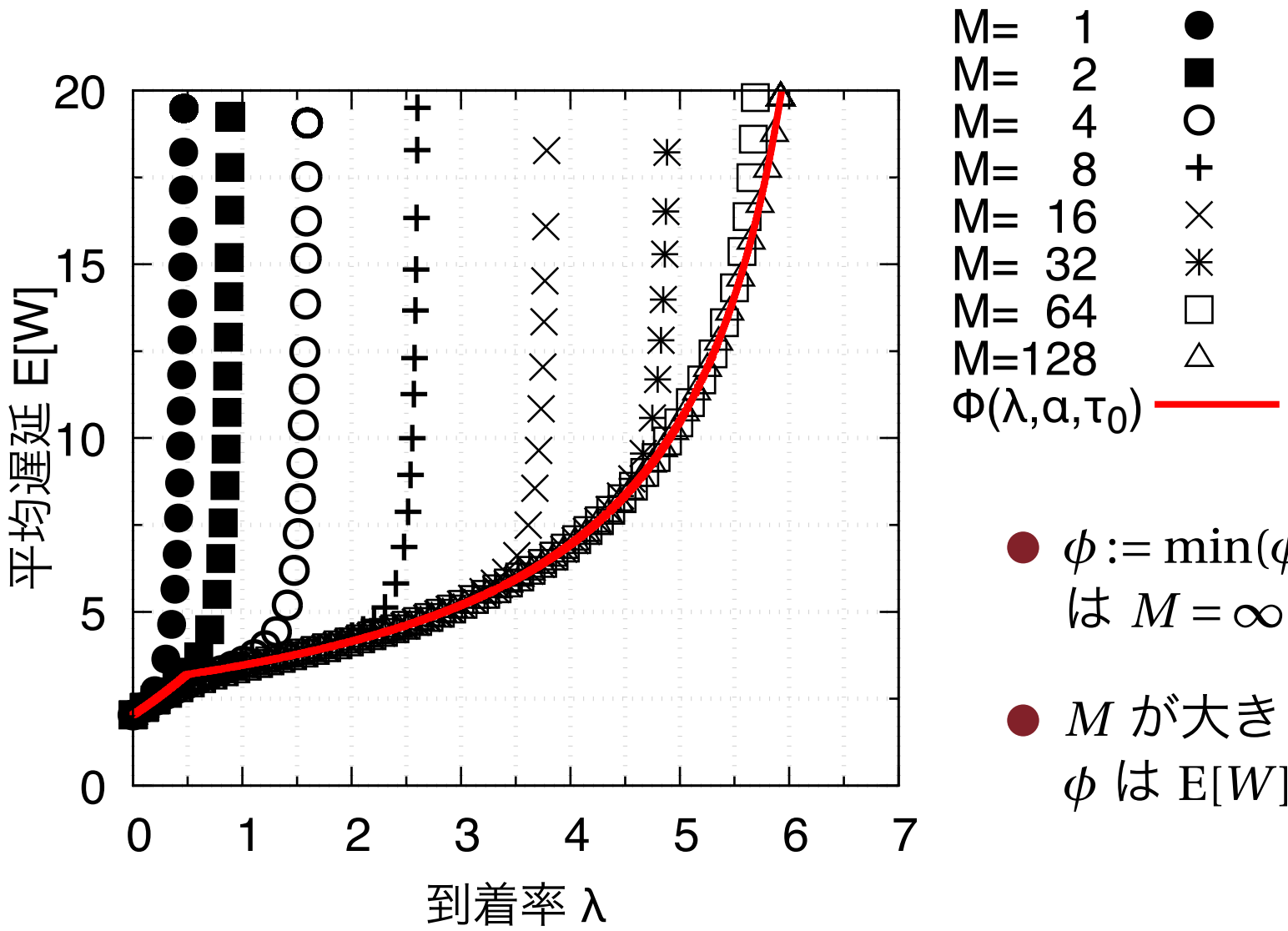
$$\mu^{[b]} = \frac{b}{\alpha b + \tau_0}$$

  - b に関して増加
- ρ の値は $\mu^{[\infty]}$ が基準

➡ 中程度の ρ でも，小さいバッチサイズに対しては過負荷

最大バッチサイズが有限の場合 (Tesla V100)

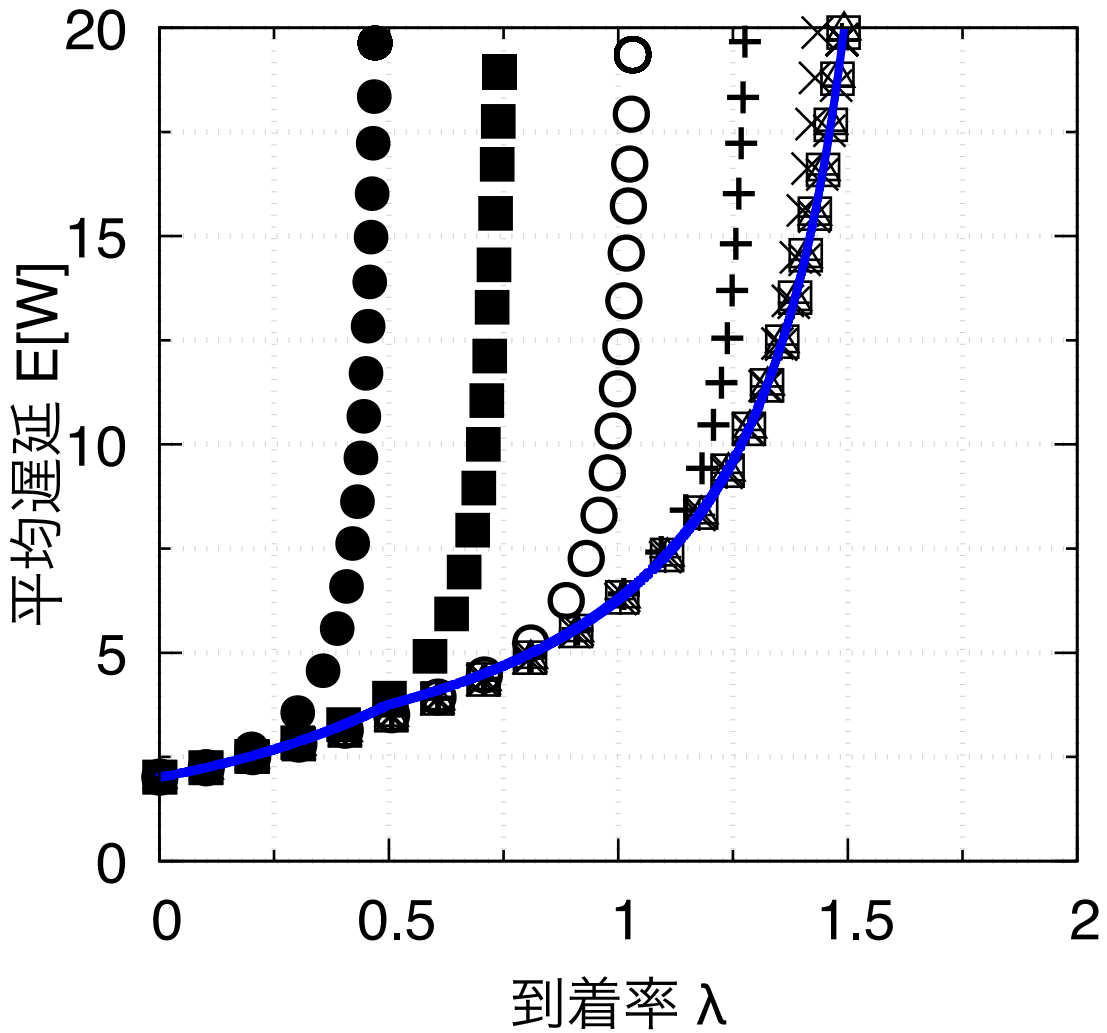
最大バッチサイズ $M<\infty$ の場合を行列解析法で計算 (Tesla V100)



- $\phi := \min(\phi_0, \phi_1)$ は $M=\infty$ での上界式
- M が大きいとき, ϕ は $\mathrm{E}[W]$ の良い近似

最大バッチサイズが有限の場合 (Tesla P4)

最大バッチサイズ $M<\infty$ の場合を行列解析法で計算 (Tesla P4)



M= 1
M= 2
M= 4
M= 8
M= 16
M= 32
M= 64
M=128
Φ(λ,α,τ0)

- Tesla V100 とほぼ同様
- より小さな M で $M=\infty$ の結果に接近

まとめ

- GPU 推論サーバを待ち行列モデル化
  - ジョブは率 λ でポワソン到着
  - サイズ b のバッチ処理時間 $H^{[b]}$ は一定値 $\tau^{[b]}$ を取る
    - $\tau^{[b]} = \alpha b + \tau_0, \quad b = 1, 2, \ldots$
- 最大バッチサイズ $M = \infty$ に対し，平均遅延 $\mathrm{E}[W]$ の上界を導出

(i) $$\mathrm{E}[W] \le \frac{\alpha + \tau_0}{2(1-\lambda\alpha)} \left(1 + 2\lambda\tau_0 + \frac{1-\lambda\tau_0}{1+\lambda\alpha}\right) =: \phi_0(\lambda, \alpha, \tau_0)$$

(ii) $$\mathrm{E}[W] \le \frac{3}{2} \cdot \frac{\tau_0}{1-\lambda\alpha} + \frac{\alpha}{2} \cdot \frac{\lambda\alpha + 2}{1-\lambda^2\alpha^2} =: \phi_1(\lambda, \alpha, \tau_0)$$

- これらの上界 (特に ϕ_1) が真値の良い近似であることを確認
- 論文中では，エネルギー効率に関しても議論