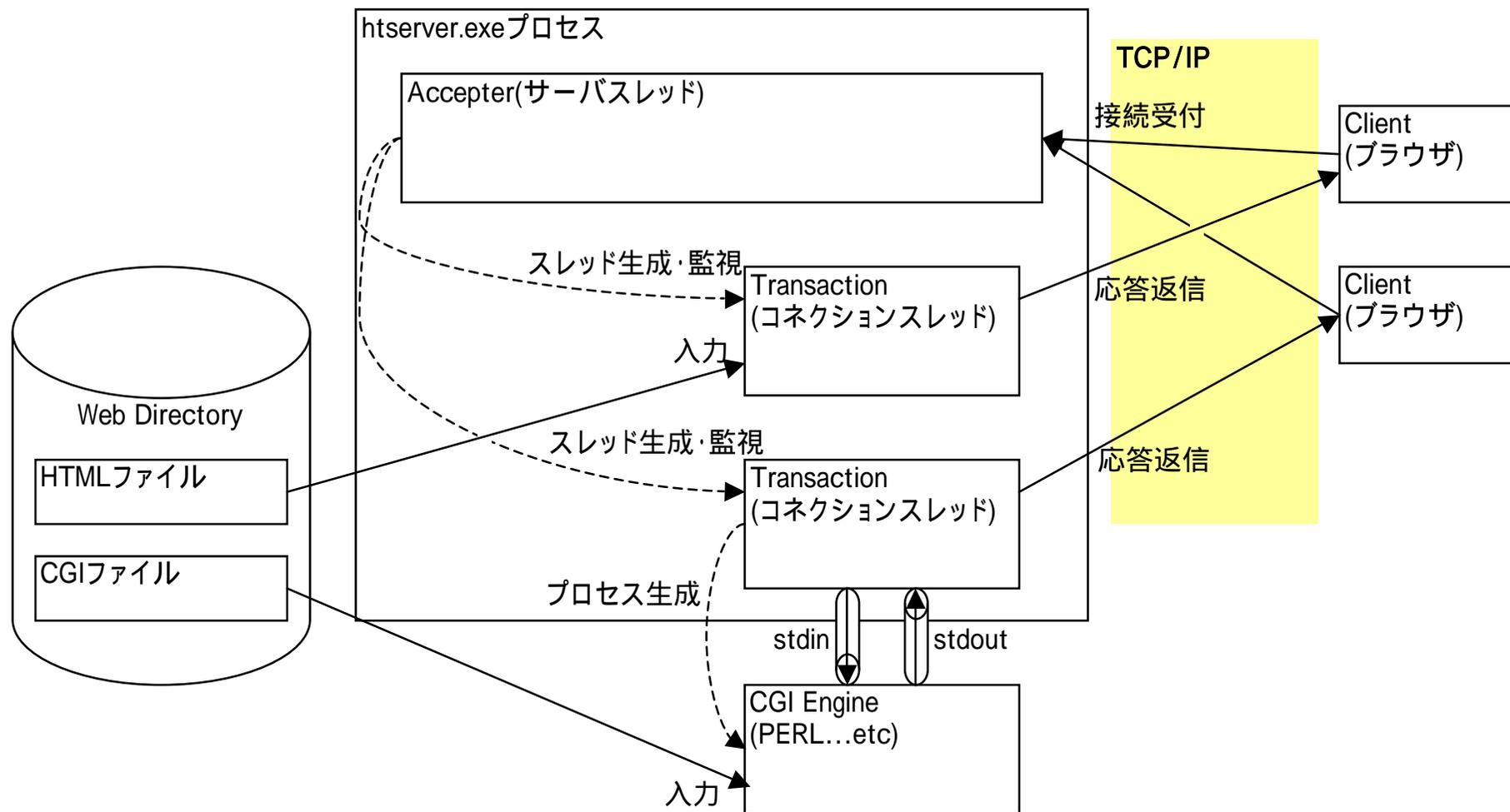


# Simple HTTP Server / スレッド・プロセス入出力構成図



## Acceptor(サーバスレッド)とTransaction(コネクションスレッド)に関して

Acceptor(サーバスレッド)は、クライアントからの接続を監視するスレッドです。クライアントからの接続を受けると、Transaction(コネクションスレッド)を生成し、以降のクライアントとの通信処理はTransactionが行います。

### 接続数の限界値に関して

接続数の限界値は、サーバの処理能力に応じて適切な値を設定すべき重要なパラメタです。接続数の限界値については、S/HTTP Serverの設定ファイルのmax-connectとsocket-backlogに影響されます。

max-connectは生成できるトランザクション数の閾値を示し、socket-backlogはサーバソケット(acceptしたソケット)に対する接続の限界値を示します。一見両者は同意のパラメタと間違えられがちですが、それぞれの意味は大きく異なります。max-connectはS/HTTP Serverの処理能力に影響し、socket-backlogはカーネルのTCP/IPドライバの受信キュー(RecvQ)の許容量です。

もしも、S/HTTP Serverを動作させるサーバマシンが数百万～数千万円もする並列プロセッサを備えた高性能なものであれば、socket-backlogにはカーネルの限界値( )を設定し、max-connectにはそれと同値を設定させるのが適切です。

しかしながら、シングルプロセッサの性能が低いマシン(多くの個人ユーザが所有するPC等)であれば、socket-backlogにはカーネルの限界値(またはそれ以上の値)を設定しても問題無いかもしれませんが、max-connectには性能に応じた控えめな値を設定すべきです。これらの設定値が不適切な場合、クライアントからのアクセスが集中した時に、プロセッサの処理能力が限界に達して、マシンがスローダウンする可能性があります。

socket-backlogに設定した値はWinSock2のlisten関数のbacklogパラメタに渡されます。この設定値がカーネルのTCP/IPキューの上限値を超えていた場合、自動的に上限値に丸められます。(詳しくはlisten関数の関数仕様をご参照ください)

## トランザクションの管理に関して

サーバスレッドは、トランザクションの稼動状況を定期的に監視しています。  
停止しているトランザクション(=クライアントへの応答処理が終わったトランザクション)は、自ら停止しますが、サーバ側に残ったハンドルはこの監視のタイミングで破棄されます。  
トランザクションの稼動状況を監視する周期についてはloop-cycle-sec(秒単位)とloop-cycle-usec(マイクロ秒単位)で指定します。  
また、session-timeout(秒単位)の期間に応答が無いトランザクションについては強制的に破棄します。  
この時、CGIプログラムが実行中であった場合、CGIプログラムのプロセス(およびその孫プロセス)は破棄されません。  
その為、CGIプログラムの実行タイムアウト時間(cgi-exe-timeout)は、session-timeoutよりも小さい値を設定すべきです。