NightStar RT Installation Guide

Version 5.0

(RedHawk™ Linux®)

concurrent REAL-TIME

0898008-5.0
November 2020

本書は当社製品を利用する社員、顧客、エンドユーザーを対象とします。
本書に含まれる情報は、本書発行時点での正確な情報ですが、予告なく変更されることがあります。
当社は、明示的、暗示的に関わらず本書に含まれる情報に対して保障できかねます。

注意事項：

本書は、Concurrent Real-Time, Inc.より発行された「NightStar RT Installation Guide」を日本語に翻訳した資料です。英文と表現が異なる文章については英文の内容が優先されます。

目次

1.0. 序文

NightStar RT Version 5.0 は、Concurrent Real-Time 社の RedHawk Linux で動作する NightStar RT の製品リリースです。

NightStar RT Version 5.0 は 2 ページの「ホスト・システム」と「ターゲット・システム」で説明されているように最新の Linux ディストリビューションでの利用が必要となります。

NightStar RT は NightProbe データ・モニター、NightSim アプリケーション・スケジューラ、NightTrace イベント・アナライザー、NightTune システムおよびアプリケーション・チューナー、NightView ソースレベル・デバッガー、Datamon モニタリング API、Shmdefine 共有メモリ・ユーティリティで構成されます。

インストール手順は NightStar 5.0 で変更されました。4 ページの「CentOS および Red Hat 向けネットワーク・インストール」および 8 ページの「Ubuntu 向けネットワーク・インストール」で説明されているようにネットワーク・インストールで使用される URL は変更されているので特に注意して下さい。

NightStar の一般的な説明については、29 ページの「NightStar RT の概要」を参照して下さい。

2.0. サポート対象システム

ホスト・システムとターゲット・システムの両方に対する NightStar RT Version 5.0 の必要条件は、以下のとおりです：

2.1. ホスト・システム

Intel と AMD の x86_84 プロセッサーおよび選定された ARM64 システムで動作する次の Linux ディストリビューションのいずれか：

- Concurrent Real-Time RedHawk Linux 7.5～8.2
- Red Hat Enterprise 7.5～8.2
- CentOS 7.5～8.2
- Ubuntu 16.04～20.04
- Debian 10

ネイティブ Intel 32-bit のサポートはもう利用できなくなりましたが、64-bit のホスト・マシーン上で 32-bit プログラムを構築および解析することは可能です。

NightStar は x86_84 ハードウェア上の 32-bit Intel プログラムをサポートします。

2.2. ターゲット・システム

RedHawk Linux は全てのターゲット・システムで必要となります。

NighStar は RedHawk Linux カーネル Version 7.5～8.2 が動作するいずれの Intel もしくは AMD の x86_64 システムをサポートします。

次の ARM64 システムをサポートします：

- NVIDIA™ Corporation の Jetson TX1, TX2, PX2-Drive, AGX-Driver, NV-Xavier, AGX-Xavier, Nano 開発キット

3.0. NightStar RT のインストール

NightStar RT をインストールするには 3 つの方法があります。

- DVD インストール
- CentOS および Red Hat 向けネットワーク・インストール
- Ubuntu 向けネットワーク・インストール

IMPORTANT

実際の**インストール前に**選択したインストール方法の全てを読むことを強く推奨します。

デスクトップ上に NightStar のアイコンがある場合、インストールを開始する前にそれらを削除して下さい。インストールが完了した後、*通常の*ユーザーとしてログインし次のコマンドを使ってアイコンを再インストールして下さい：

```
/usr/lib/NightStar/bin/install_icons
```

本項の残りのコマンドは root ユーザーとして実行して下さい。

3.1. DVD インストール

NightStar RT Installation DVD を使って NightStar RT をインストールするには：

- DVD-ROM ドライブに *NightStar RT Installation DVD* を挿入して下さい。新しいシステムでは次の場所のいずれかに自動でマウントされます：

  - **/media/NightStar-RT-5.0**
  - **/run/media/{user-name}/NightStar-RT-5.0**

- DVD が自動マウントしない場合、次のような方法で DVD-ROM ドライブをマウントして下さい：

  ```
  [ -d /mnt/cdrom ] || mkdir /mnt/cdrom;
  mount -t iso9660 -o ro /dev/sr0 /mnt/cdrom
  ```

  お手持ちの DVD デバイスが**/dev/sr0** 以外である可能性があります。

- 現在の作業ディレクトリを DVD がマウントされているディレクトリに変更し、root ユーザーとして次のスクリプトを起動して下さい。：

  ```
  ./install-nstar
  ```

x86_64 システムでのインストール中、32-bit の API とデバッグ機能をインストールするかどうかを尋ねられます。これは完全なオプションです。この時点で選択しない場合、後で install-nstar スクリプトを再実行することでいつでもそれらをインストーすることが可能です。

NOTE

前述のインストール手順はお手持ちのシステムに NightStar 製品全体をインストールします。組み込みシステムに関しては、NightStar の一部のターゲット側をインストールし、組み込みシステムを対象とする他のシステムから全ての GUI 操作を行うのが良い可能性があります。ターゲット側分だけをインストールするには、現在の作業ディレクトリをマウントされた DVD に変更して、

```
./install-nstar-server
```

install-nstar スクリプトの代わりに上記スクリプトを起動して下さい。

3.2. CentOS および Red Hat 向けネットーワーク・インストール

3.2.1. 最初に NightStar を削除

お手持ちのシステム内に存在する NightStar インストールを最初にいずれも削除して下さい。このために **`remove-nstar`** スクリプトが提供されており、DVD のベース・ディレクトリおよびオンラインの https://redhawk.concurrent-rt.com/NightStar/remove-nstar で見つけることが可能です。root ユーザーとして次のスクリプトを実行して下さい：

```
bash ./remove-nstar
```

本スクリプトはシステムにいくつかの NightStar パッケージを残す可能性があります。これが発生する場合、何故そしてどのパッケージが留まるのかが分かります。これは問題ではないのでこの手順から継続することは可能です。

`remove-nstar` スクリプトの md5sum チェック・サムは 4f0b36e79c533239d7f4758dbee68f98 となります。

3.2.2. NUU, YUM, DNF 経由でのインストール

ネットワーク・インストールはいずれかのコマンド(**`nuu`**, **`yum`**, **`dnf`**)を使って https://redhawk.concurrent-rt.com を介して達成されます。nuu は Concurrent Real-Time の Network Update and installation Utility となります。

リポジトリへのアクセスに必要な **`redhawk.concurrent-rt.com`** のログインとパスワードを管理しているので、**`nuu`** の利用を推奨します。

システムに NUU がインストールされていないけど利用したい場合、次の URL にアクセスし NUU のインストールと設定の指示に従ってから次のセクションに戻って下さい：

http://redhawk.concurrent-rt.com/network/yum.html#NUU

3.2.2.1. YUM 構成ファイル

使用するコマンドに関わらず、次の構成ファイルはシステムに存在する必要があります：

```
/etc/yum.repos.d/ccur-nstar-rt.repo
```

ファイルが存在しない場合は生成して下さい。次の記述に従い、色付きの斜体のテキストを固有の情報に置き換えて適切な内容になるように修正して下さい。

```
[ccur-nstar-rt]
name=NightStar RT
baseurl=https://redhawk.concurrent-
rt.com/rhel/Login=login/Password=password/NightStar/RT/version/distro/$basearch
gpgcheck=0
enabled=1
```

IMPORTANT

ファイル内の重要な行は「baseurl」で始まり「$basearch」で終わる行です。これはスペースも改行もない(本書でどのように見えているかに関係なく)単一の行です。

上記の太字 ***green*** 部は、Concurrent Real-Time から割り当てられた redhawk.concurrent-rt.com の ***Login ID*** と ***Password*** に置き換える必要があります。この情報はシステムまたはソフトウェアと一緒に出荷されたカバー・レターの中に含まれています。Login ID は通常は LA または LR で始まり続いて 5〜7 桁の数字の Site ID でもあります。もしこれらの値がすぐに見つからない場合、51 ページの「ソフトウェアの直接サポート」のセクションに示されているようにコンカレント日本に連絡して下さい。これらの情報を埋めることが出来るようになるまで、ファイルの最後の行の「enabled」変数を 0 に設定することが可能です。

赤い斜体の太字の ***version*** コンポーネントは NightStar のバージョンで次のいずれかである必要があります：

4.1, ***4.2***, ***4.3***, ***4.4***, ***4.5***, ***4.6***, ***4.8***, ***5.0*** または ***current***

NightStar の本リリースでは ***5.0*** を使用して下さい。あるいは ***current*** を使用して下さい。これは最新の製品リリースのための代用語です。NightStar の最新の製品リリース(例えば NightStar 5.1 がリリースされた場合)に常にアクセスできるように値に ***current*** を使用することを推奨します。

NOTE

以前のリリースでは、値として ***RedHawk*** が ***current*** の代わりに使用されていました。(前者は廃止される予定です)

NightStar 5.0 では、ベース URL は ***distro*** コンポーネント(Linux ディストリビューションの略語)を含んでいる必要があります。お手持ちのシステムのディストリビューションを最もよく表している次の値のいずれかを使用して下さい：

Linux ディストリビューションの略語

ディストリビューション	略語
CentOS	***rhel7***, ***rhel8***
RHEL	***rhel7***, ***rhel8***
Ubuntu	***ubu16***, ***ubu18***, ***ubu19***, ***ubu20***
Debian	***deb10***

IMPORTANT

5.0 より前の NightStar のバージョンを指定する場合、/***distro*** コンポーネントを完全に省略して下さい。

通常、$basearch を他の値に置き換える必要はありません。このパスは$basearch で終える必要があり、実行時に yum ユーティリティによって自動的に設定される yum の変数です。

構成ファイルは次のようになります：

```
[ccur-nstar-rt]
name=NightStar RT
baseurl=https://redhawk.concurrent-
rt.com/rhel/Login=LA12345/Password=secret/NightStar/RT/5.0/rhel8/$basearch
gpgcheck=0
enabled=1
```

IMPORTANT

ファイル内の重要な行は「baseurl」で始まり「$basearch」で終わる行です。これはスペースも改行もない(本書でどのように見えているかに関係なく)単一の行です。

3.2.2.2. NUU 経由でのインストール

NUU がインストールされたら、次のように NUU を起動して本リリースをインストールすることが可能です：

```
/usr/bin/nuu --disablerepo=ccur-* --enablerepo=ccur-nstar-rt
```

Concurrent Real-Time 製品リポジトリからインストールまたはアップグレードする NUU を利用したことがない場合、ログインおよびパスワード情報を求められる可能性があります。

NightStar RT がまだインストールされていない場合、NUU のモードを規定値は *Updateable* なので *Installable* に変更する必要があります。この設定はメイン・ウィンドウの右上にあるドロップダウン・メニューから選択することが可能です。設定するとパッケージ内のリストで **`ccur-NightStar-RT-RedHawk`** を見つけることが可能となります。インストールするパッケージを選択して **`Apply`** ボタンをクリックして下さい。そのパッケージの依存関係は、オプションの 32-bit サポートを除いて NightStar RT 製品全体をインストールします。32-bit サポートのインストールは以下で個別に対処します。

システムに NightStar RT が既にインストールされている場合、古いパッケージは画面上の更新可能リストに表示されます。更新するにはそれらを選択して **`Apply`** ボタンをクリックして下さい。

IMPORTANT

更新時は注意して下さい。NUU は基本ディストリビューションおよび NightStar からソフトウェアをインストールします。意図するものではない限りシステム全体を誤って更新しないようにして下さい。NUU の **`Apply`** を押下した後、更新するパッケージを正確に表示するダイアログがポップアップされます。開始する前にそのリストを慎重に調べて下さい。全ての Concurrent Real-Time のパッケージは接頭語「ccur-」で始まります。

http://redhawk.concurrent-rt.com/network/yum.html#NUU から入手できる NUU ダウンロード・キットに含まれる QuickStart.pdf の資料は、NUU の使用方法をより詳細に説明しています。

32-bit NightStar サポートをインストールしたい場合、NUU を使用して次の RPM を明示的にインストールして下さい：

```
ccur-nstar-32bit-support
```

32-bit API および x86_64 システム上でのデバッグの完全サポートを得るために本 RPM は 5 つの追加 RPM をインストールします。

3.2.3. YUM および DNF 経由でのインストール

最新の RPM ベースのシステムでは、**yum** コマンドは **dnf** コマンドに置き換えられました。以下のインストール・コマンドについては、システムにインストールされているいずれのツールを使用して下さい。

IMPORTANT

NightStar が既にインストールされている場合、続行する前に削除する必要があります。4 ページの「最初に NightStar を削除」を参照して下さい。

3.2.3.1. インストール

NightStar がインストールされていないシステムに NightStar RT をインストールするには、次のコマンドを使用して下さい：

```
yum --disablerepo=ccur-* --enablerepo=ccur-nstar-rt install \
ccur-NightStar-RT-RedHawk
```

3.2.3.2. 将来的に NightStar を更新

Concurrent は必要に応じて NightStar 5.0 のアップデートを提供します。NightStar 5.0 が既にインストールされているシステムの NightStar を更新するには次のコマンドを使用して下さい：

```
yum --disablerepo=ccur-* --enablerepo=ccur-nstar-rt update \
ccur-NightStar-RT-RedHawk 'ccur-n[^v]*' ccur-qt5 \
ccur-datamon ccur-shmdefine
```

3.2.3.3. 更なるネットワーク・インストールの問題

NOTE

次の廃止されたパッケージを更新またはインストールしないで下さい：ccur-NightView, ccur-NightView-docs-rt, ccur-Nviewp, ccur-Nview-ptrace, ccur-ntunecommon, ccur-ntuneserv, ccur-ntracelog, ccur-ntraceapi, ccur-nsimserver, ccur-nprobeserv, ccur-ntrace-api-libs, ccur-ntrace-java-api-libs, ccur-nprobe-api-libs

インストール後、上述の更新コマンドを使って対話形式にシステムを最新の状態にしておくことが可能です。恐らくシステムを毎晩更新するために **cron** ジョブをインストールすることも可能です。例：

```
1 0 * * * /usr/bin/yum -y --disablerepo=* \
--enablerepo=ccur-nstar-rt update ccur-NightStar-RT-RedHawk \
ccur-n[^v]* ccur-qt5 ccur-datamon ccur-shmdefine
```

32-bit NightStar サポートをインストールする場合、次のコマンドを使って **ccur-nstar-32bit-support** の RPM を明示的にインストールして下さい。

```
yum install --disablerepo=ccur-* --enablerepo=ccur-nstar-rt \
ccur-nstar-32bit-support
```

32-bit API および x86_64 システム上でのデバッグの完全サポートを得るために本 RPM は 5 つの追加 RPM をインストールします。

3.3. Ubuntu 向けネットーワーク・インストール

IMPORTANT

NightStar が既にインストールされている場合、続行する前に削除する必要があります。4 ページの「最初に NightStar を削除」を参照して下さい。

Ubuntu および Debian システムへの NightStar のインストールは、これらのディストリビューションにおいて標準的なインストール手法である **apt** を使用して簡単に完了します。

Concurrent Real-Time の公開キーを次に示すように **apt-key** コマンドを使って apt キー・リングに追加する必要があります。公開キー・ファイルはインストール DVD にも存在することに留意して下さい。

```
wget http://redhawk.concurrent-rt.com/network/ccur-public-keys
apt-key add ccur-public-keys
```

3.3.1. APT リポジトリ構成ファイル

NightStar RT リポジトリ定義の URL は **redhawk.concurrent-rt.com** のログイン情報および特定の Linux ディストリビューションにカスタマイズする必要があります。

次の内容で斜体の色付きの文字をシステム固有の情報に置換して**/etc/apt/sources.list.d** ディレクトリに **ccur-nstar.list** という名前のファイルを生成して下さい：

```
deb http://redhawk.concurrent-rt.com/ubuntu/login/password/nightstar \
5.0 rt distro
```

IMPORTANT

「deb」で始まり「*distro*」で終わるリポジトリ定義の行は、上記がどのように表示されるかに関係なく 1 行である必要があります。バックスラッシュ文字を削除し、5 個のコンポーネントの間は 1 個のスペースにして下さい。

login と ***password*** 欄は redhawk.concurrent-rt.com 用のログインとパスワードです。これらの値は、システムまたはソフトウェア購入時に受け取ったカバー・レターに記載されています。***login*** 部分はサイト ID とも呼ばれています。サイト ID は通常 LA または LR で始まり続いて 5〜7 桁の数字となります。

この行は ***distro*** コンポーネント(Linux ディストリビューションの略語)を含む必要があります。お手持ちのシステムのディストリビューションを最もよく表している次の値のいずれかを使用して下さい：

Linux ディストリビューションの略語

ディストリビューション	略語
Ubuntu	***ubu16***, ***ubu18***, ***ubu19***, ***ubu20***
Debian	***deb10***

IMPORTANT

5.0 より前の NightStar のバージョンを指定する場合、***distro*** コンポーネントを完全に省略して下さい。

一般的な deb 定義行は次のようになります：

```
deb http://redhawk.concurrent-rt.com/ubuntu/LA12345/secret/nightstar \
5.0 rt ubu20
```

全て 1 行､バックスラッシュ文字(\)なしで 5 個のコンポーネントを 1 つのスペースで区切ります。

3.3.2. APT インストール・コマンド

IMPORTANT

NightStar が既にインストールされている場合、続行する前に削除する必要があります。4 ページの「最初に NightStar を削除」を参照して下さい。

apt コマンドはリポジトリのソース・リストの再読み込みを強制します。次のコマンドを実行して下さい(コマンドは本手順の一部としてパッケージを変更することはないので、update という言葉については心配しないで下さい)：

```
apt-get update
```

NightStar がまだインストールされていない場合は次のコマンドを実行して下さい

```
apt-get install ccur-nightstar-rt
```

一方、システムに NightStar が既にインストールされている場合、次に示すように特有のコマンドを実行する必要があります：

```
apt-get install “ccur-n[^v]*” ccur-qt5 ccur-datamon ccur-shmdefine
```

3.3.3. Ubuntu での NightStar 32-bit サポート

NightStar は 64-bit システム上での 32-bit プログラムのビルド、実行、デバッグを引き続きサポートします。

x86_64 システムに 32-bit パッケージをインストールするには、サポートするパッケージのアーキテクチャのリストに i386 を追加する必要があります。次のコマンドを実行することでお手持ちのシステムが既に i386 パッケージのインストールを許可しているかどうかを確認することが可能です：

```
dpkg --print-foreign-architectures
```

アーキテクチャのリストに i386 を追加するには次のコマンドを実行して下さい：

```
dpkg --add-architecture i386
```

システムが i386 をサポートしていることを確認したら、次の 2 つのコマンドを実行して下さい：

```
apt-get update
apt-get install ccur-nstar-32bit-support
```

最初のコマンド **`apt-get update`** はシステムにいずれのパッケージもインストールしません。定義されている全てのレポジトリに関するデータベース情報だけを更新します。

2 番目のコマンドは ccur-nstar-32bit-support パッケージと共に 5 個の依存パッケージをもインストールします。

4.0. NightStar RT 5.0 の変更点

4.1. 新機能

NightStar RT 5.0 の主な変更点は次を含みます：

- RedHawk 8.2 をサポート
- Qt Version 5.13 を使用 (Version 4.2.2 からアップグレード)
- NVIDIA の CUDA 11 リリースをサポート
- NightView で C++例外処理をサポート
- 32-bit サポートの簡易化
- 新しい Linux ディストリビューション(RHEL/CentOS 8.2 と Ubuntu 20.04)をサポート
- 更なる ARM64 NVIDIA 開発キットを認定
- AMD64 AVX512 命令をサポート

4.2. 最新機能

NightStar 5.0 は全てのバグ修正および NightStar 4.8 で追加された新しい機能も含まれています。

4.3. NightView の変更点

- NightView はすでに CUDA 11 をサポート
- NightView は C++例外オブジェクトのキャッチをサポート(11 ページの「C++例外処理」を参照して下さい)
- NightView はメモリ内の生の値が様々なサイズや書式(16 進バイト、ワード、ASCII 等)で表示されるメモリ編集パネルが特徴です。本パネル内で文字列や 16 進数値を検索することが可能です。値の変更も可能です。詳細については *NightView User's Guide* の「バイナリ・ビュワー・パネル」を参照して下さい。
- NightView は「System V アプリケーション・バイナリ・インターフェース Intel386 アーキテクチャ・プロセッサ・サプリメント Version 1.0 (2015 年 2 月 3 日)」に記述されている 16-byte および 32-byte i386 スタック・アライメント要件を固持しています
- I/O、停止理由、コマンド等を分けるために色付けされたメッセージ・パネル

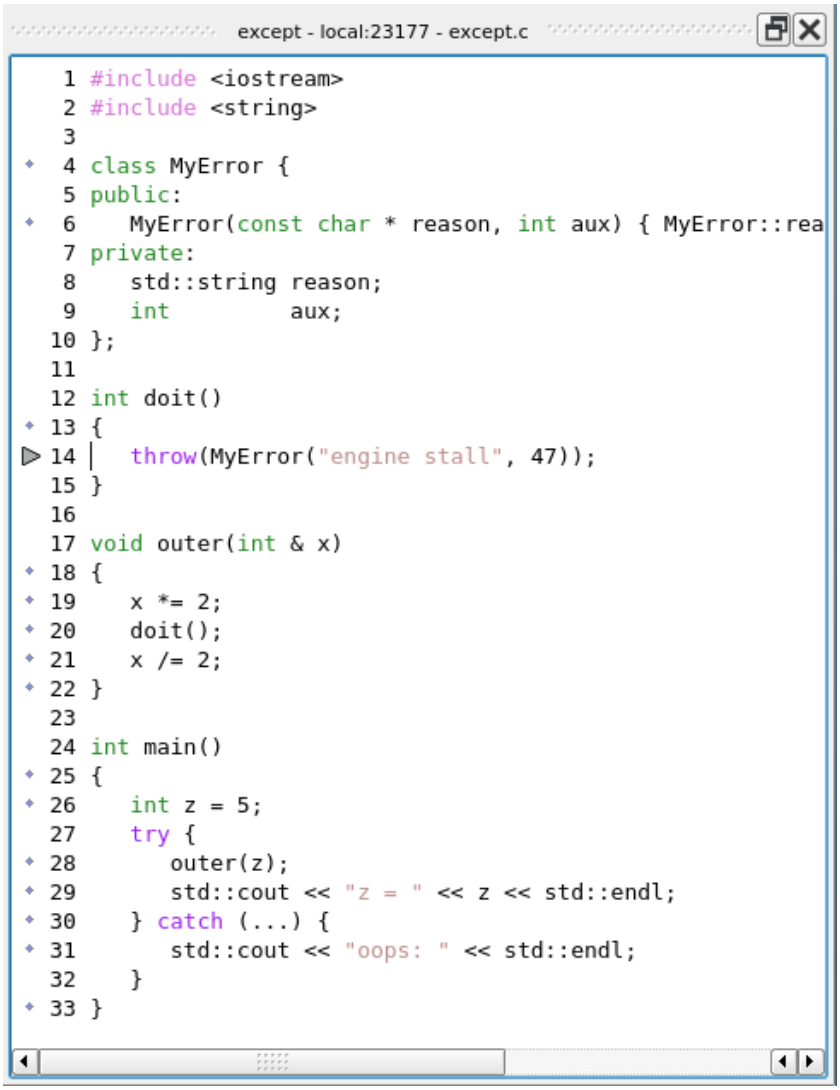
4.3.1. C++例外処理

NightView は C++例外をキャッチできるようになりました。

例外が投げられたのを検出するように NightView に指示するには **`handle`** コマンドを使用します:

```
handle/c * stop print
```

このコマンドは、いずれの種類の C++例外オブジェクトが投げられた時に NightView にプロセスを停止するように指示します。次図で示すように **`handle`** コマンドの後にプロセスを再開したとすると、NightView は 14 行目でプロセスを停止します。



図： 例外が投げられて停止

例外が投げられたので、NightView は 14 行目でプログラムの実行を停止しました。

NightView は次のテキストをメッセージ・パネルに出力します：

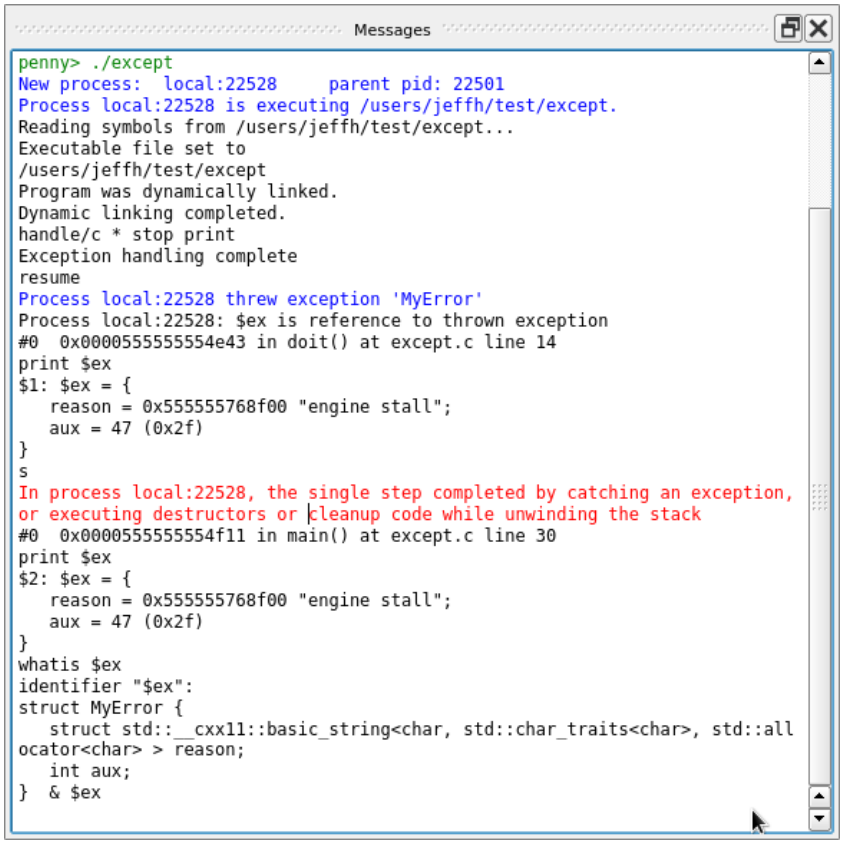
```
Process local:22528: threw exception ‘MyError’
Process local:22528: $ex is a reference to thrown exception
```

$ex は投げられるオブジェクトを表す NightView の便利な変数であることを理解して下さい。本変数は例外がキャッチされた位置のコンテキスト内でも有効です。コマンド **step** で入る場合、例外がキャッチされた場所である 30 行目になります。この時点でも投げられたオブジェクトに関する詳細な情報を得るために**$ex** を評価することが可能です。

```
print $ex
$1: $ex = {
     reason = 0x5214648 “engine stall”;
```

```
        aux = 47 (0x2f)
}
```

上述のデバッグ・セッション全体を次図で示します。



図： ハンドラーに対する NightView セッションの手順

4.4. NightTrace の変更点

NightTrace の最近の変更の一部は次を含んでいます：

- ユーザー・イベントがデフォルトでカーネル・タイムライン上に表示
- NightTrace は Python のロギング API を提供(15 ページの「Python における NightTrace のバインド」を参照して下さい)

- カスタム・カーネル・タイムラインは 8 個以上の CPU を搭載するシステムで有用となる可能性のあるより多くのオプションを提供(16 ページの「カスタム・カーネル・タイムライン・オプション」を参照して下さい)
- カーネルとユーザーのトレース・データの両方を使用する場合における PID 追跡の改善(17 ページの「カーネル/ユーザー・トレース・データに対する PID 追跡の向上」を参照して下さい)

4.4.1. Python における NightTrace のバインド

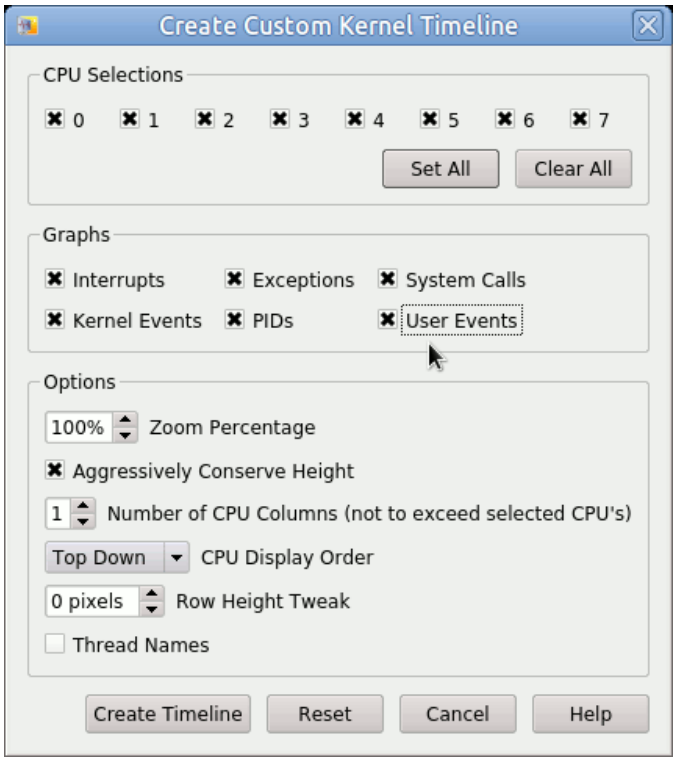
Ada, C 言語、Fortran, Java のバインディングに加え、NightTrace はトレース・イベントのロギング用に Python インターフェースを提供するようになりました。

簡単な使用例は次のようになります：

```
import sys
sys.path.append("/usr/lib/NightTrace/python/python2")
import ntrace
tr=ntrace.log("/tmp/user-data")
tr.Event(47)
tr.Event("pi", 3.14159)
tr.Event("init", "some interesting string")
tr.Event("my_list", [1,2,3,4])
tr.Event("my_tuple",(4.5,3.2))
```

API の詳細については**/usr/lib/NightTrace/python/python2/ntrace.py**、更なる使用例については**/usr/lib/NightTrace/python/python2/examples** を参照して下さい。

4.4.2. カスタム・カーネル・タイムライン・オプション



図： カスタム・カーネル・タイムラインの生成ダイアログ

Graphs グループ領域において、新たなオプション「**[x] User Events**」が追加されました。本オプション・グループ領域では、4 つの新しいオプションが追加されました。これらは生成しようとしているカーネル・タイムラインの体裁を制御します。

User Events

オンにした場合、ユーザー・イベントがユーザー・タイムライン上だけでなくカーネル・スタンザにも表示されます。ユーザー・イベントを表示する短い行が各 CPU スタンザの上部に現れます。

Aggressively Conserve Height

オンにした場合、フォントの上部周辺のかなりのピクセルが切り取られたとしても NightTrace は行のピクセル高を減らすことで行の高さを積極的に計算します。

縦方向のスペースが重要ではない場合、本ボックスはオフにして下さい。

Number of CPU Columns

NightTrace は CPU の並ぶ列を最大 4 つまでサポートします。これは表示されるタイムラインの幅を縮小しますが、ワイド・モニターを使用すると役立つ可能性があります。

この値は表示する CPU 数と CPU 閾値の設定をベースにして NightTrace に最初に設定されます。値を 1 に設定すると常に 1 列のタイムラインにすることが可能です。

17 ページの「Row Height Tweak」で説明しているように閾値の設定を変更することも可能です。

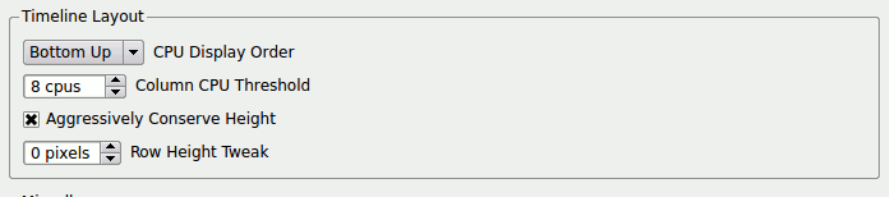
CPU Display Order

列内の CPU の論理的な順序を制御して下さい。Bottom Up は CPU 0 を列の最下部に配置しますが、従来の NightTrace の方式は CPU 0 を列の最上部に配置します。

Bottom Up は最初のチップ上の CPU に集中したい状況では望ましい可能性がある一方、最も高い番号の CPU は画面から外れる可能性がありますので上にスクロールする必要があります。

Row Height Tweak

本設定は行の高さに関する追加の制御を提供します。ここでは NightTrace が選択した値に対しピクセルを加算または減算することが可能です。この値は NightTrace が他のこと全てを考慮した後に適用されます。



図： 設定ダイアログのタイムライン部分

ここでは Custom Kernel Timeline ダイアログで示されるのと似たような設定が見えます。これらの設定はデフォルトのカーネル･タイムラインが生成される時に NightTrace により閲覧されます。設定をその後のセッションで使用するために保存することが可能ですが、試すことが可能なので現在のセッション中に設定を変更することも可能です。

Preferences ダイアログは File メニューで見つけられます。

4.4.3. カーネル/ユーザー・トレース・データに対する PID 追跡の向上

プロセス ID に加えプログラム名称を提供する機能はトレース・データの可読性を大幅に高めます。

これまでは、プロセス名称はトレース・イベントの大半で利用可能でしたが、PID のみが利用可能であった多くのケースがありました。このような状況は、コンテキスト・スイッチが発生する前にカーネル・トレースを開始した時に確認されたプロセス、同様にカーネル・トレースを開始した後に起動されたけどカーネル・トレースを停止する前に終了したプロセスを含んでいました。

本リリースでは、NightTrace はプロセス名称の決定を強化しました。殆ど全てのイベントはプロセス名称と共に PID 値を持つようになりました。これはカーネル・イベントに伴って発生した時のユーザー・イベントを含んでいます。

しかしながら、いくつか制限があります。多くのトレース・データは消費されるため、プロセス名称の決定の大半は遡及的に発生します。

ファイルからトレース・データを見るのに ntrace を使用している間、少数のプロセス名称のないイベントに遭遇するはずです。

しかしながら、トレース・イベントのストリーミング時は、初めはプロセス名称が利用可能ではない場合がありますが、多くのトレース・データが消費されるのでそれらはその後に特定されるようになります。

NightTrace 解析 API にも同様の制限があります。完全なプロセス名称の決定は、トレースの最後からデータが消費されるまでは完了することが出来ません。従って、全てのプロセス名称が決定されるのを確実にしたい場合、次のようなコードを使用する必要があります：

```
#include <ntrace_analysis.h>
...
tr t = tr_init()
...
-- Ensure the end of trace data has been seen
while (tr_next_event(t)!=TR_EOF){}
tr_seek(t,0);
...
-- Now install your conditions and states and callback
-- functions and start processing for real.
...
```

これは処理に関しては無駄が多いのですが効果的です。より洗練されたソリューションが将来検討されます。

4.5. NightSim の変更点

グラフィカル・インターフェースへの変更は小規模ですが言及する価値はあります。

- Monitor ページの表示中にスケジューラーが停止した時に NightSim はより良い通知を提供するようになりました。この表示は外部の影響(例えば、FBS に関するプロセスの 1 つがデバッガ内のブレーク・ポイントにヒットした、またはスケジューラーが他のパーティによって停止された)が原因でスケジューラーが停止した時にユーザーの混乱を回避するのに役立ちます。
- Monitor ページ上のパフォーマンス監視履歴を消去するために追加されたハンディ・アイコンとキーボード・ショートカット(Ctrl+D)

4.6. NightTune

NightTune 4.0 は CUDA 9.2, 10, 11 をサポートするようになりました。

5.0. NightStar のリモート・デバッグ

通常、NightStar はターゲット・システムがツールを実行するホスト・システムとは異なる場合、**openssh-server** がターゲット・システムにインストールされている必要があります。

リモート・デバッグ時、NightView はターゲット・システムと接続するのに **ssh** を使用します。複数の接続が必要で、各リモート・セッションで最大 3 つとなります。そのため、NightView は同じターゲット・システムの認証方法を複数回要求する可能性があります。複数認証要求を回避する最善の方法は、事前にロードされた認証キーで **ssh** エージェント(**ssh-agent(1)**を参照)を使うことです。正しく構成されていれば、NightView はターゲット・システムがエージェント提供のキーを拒否した場合にのみ認証情報を要求します。

もしくは **ssh** 制御パス転送を使用することです。これはアクティブな **ssh** 接続を通してポート転送を提供します。最近の一部の Linux ディストリビューションは制御パス転送を使用時にポート転送の設定に問題があるため、現在 NightView は本機能を控えめに使用しています (例えば、**ssh -M -o file** *target* などの最初の **ssh** 接続、それに続く接続マスターを参照する **ssh** ポート転送コマンド)。

NightView セッションでデバッグ・フラグを設定することにより NightView で本機能を完全に有効にすることが可能です。

```
set-debug use-control-path-forwarding=1
```

完全に有効にするとターゲット・システムへの 2 番目または 3 番目の **ssh** 接続を生成しようとしてハングする可能性があります。これまでの経験でハングせずに実際に接続している場合は正常に動作します。

制御パス転送を完全に無効にするには、次の NightView コマンドを使用して下さい：

```
set-debug no_ssh_port_forwarding=0
```

事前にロードされた認証キーで **ssh** エージェント(**ssh-agent(1)**を参照)を使用することを推奨します。正しく設定した場合、ポート転送方式の使用時にもかかわらず認証情報を対話形式で提供せざるを得ない状態を回避することが可能となります。

同様に NightProbe, NightSim, NightTrace, NightTune はホストとターゲット・システム間の通信のためのメカニズムとして ssh を使用します。

5.1. クロス・デバッグ

異なるアーキテクチャのホストとターゲット・システム間でのリモート・デバッグ・セッションの状態に対してクロス・デバッグという用語を使用します。クロス・デバッグはホスト・システムに **ccur-nview-*-support**、32-bit アプリケーションをデバッグしたいターゲットの x86_64 システムに **ccur-nview-i386-target** を追加でインストールする必要があります。

x86_64 の NightView は i386 と aarch64 ターゲット・システムへのクロス・デバッグをサポートします。これはホストに **ccur-nview-i386-support** または **ccur-nview-aarch64-support**、ターゲット・システムに **ccur-nview-i386-target.i386** がインストールされている必要があります。

クロス・デバッグする場合、ターゲットのアーキテクチャのコードを生成可能なコンパイラーが必要です。i386 については x86_64 のコンパイラーが-m32 オプションを使用するだけでサポートするので状況は簡単です。

x86_64 から arm64 をクロス・デバッグする場合、完全なコンパイル・ツール・チェーンが必要です。見つけるのは難しくはありませんが、追加のソフトウェア・パッケージをホスト・システムにインストールする必要があります。

6.0. ライセンス・キーの入手

NightStar RT Version 5.0 ソフトウェアは、NightStar RT の以前のバージョンと同じライセンス・マネージャーおよびライセンス機能を使用します。既に NightStar RT を使用している場合は、新しいライセンスを入手する必要はありません。

恒久的なライセンス・キー

- NightStar RT を購入している場合、次の URL で恒久的なライセンスを入手することが可能です：

  http://concurrent-rt/custom-support

  お手持ちのサイト ID、email アドレス、製品インストール時の表示されたシステム ID 番号が必要となります。ライセンス・キーをインストールするシステム上で次のコマンドを実行することで、再度システム ID 番号を取得することが可能です。

  ```
  /usr/bin/nslm_admin --code
  ```

NightStar License Manager (NSLM)に関する詳細な情報については、39 ページの「NightStar RT ライセンシング」を参照して下さい。

6.1. NightStar RT の削除

NightStar RT を削除するには、root ユーザーとして **`remove-nstar`** スクリプトを実行する必要があります。

本スクリプトはインストール DVD および Concurrent Real-Time のネットワーク https://redhawk.concurrent-rt.com/NightStar/remove-nstar で入手することが可能です。

`remove-nstar` スクリプトの md5sum チェック・サムは 4f0b36e79c533239d7f4758dbee68f98 となります。

7.0. 資料

以下の表に Concurrent Real-Time から入手可能な NightStar RT 5.0 の資料を示します。

NightStar RT Version 5.0 資料

マニュアル名称	文書番号
NightStar RT Installation Guide (Version 5.0)	0898008-5.0
NightStar RT Tutorial (Version 5.0)	0898009-170
NightProbe User's Guide (Version 4.4)	0898465-120
NightSim User's Guide (Version 4.5)	0898480-080
NightTrace User's Guide (Version 7.6)	0898398-200
NightTune User's Guide (Version 3.8)	0898515-050
NightView User's Guide (Version 7.7)	0898395-400
Data Monitoring Reference Manual	0898493-030
Quick Reference for shmdefine	0898010-060

本 *Installation Guide* のバージョンの PDF および上記の他の全ての資料が次の場所で見つけることが可能です：

- *NightStar RT Installation DVD* の **`documentation`** ディレクトリ
- *オンライン*で http://redhawk.concurrent-rt.com/docs
- インストール後の**`/usr/share/doc/NightStar/pdf`** ディレクトリ

インストール後、これら全ての資料のバージョンの HTML に次を介してアクセス可能です：

- いずれの NightStar ツールの Help メニュー
- **`/usr/bin/nhelp`** コマンド
- **`/usr/share/doc/NightStar/html`** ディレクトリ

8.0. NightStar RT GUI の機能

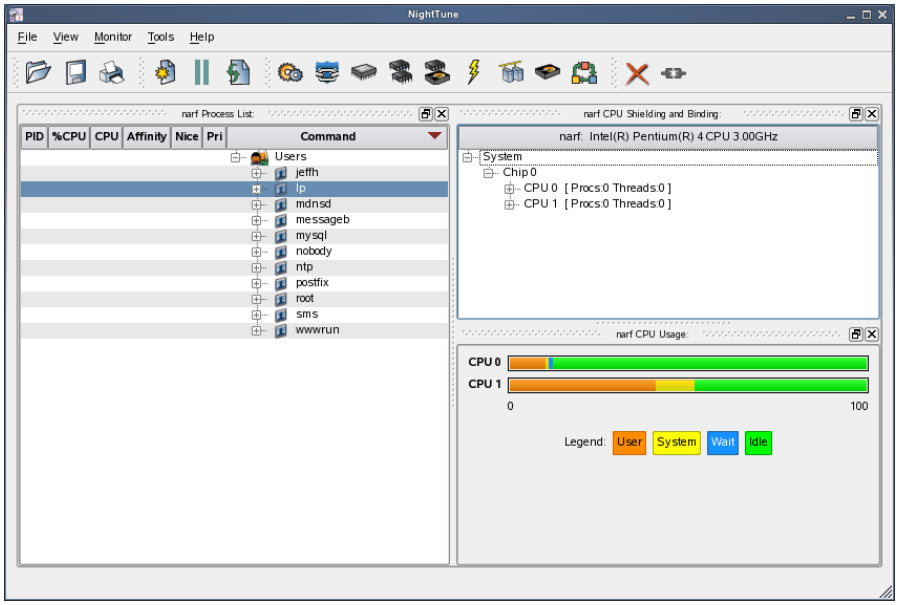
NightStar RT ツールのグラフィカル・ユーザー・インターフェースにおける一般的な機能の一部は次のとおり：

- 移動可能およびサイズ変更可能なパネル
- タブ化ページ
- コンテキスト・メニュー

8.1. 移動可能およびサイズ変更可能なパネル

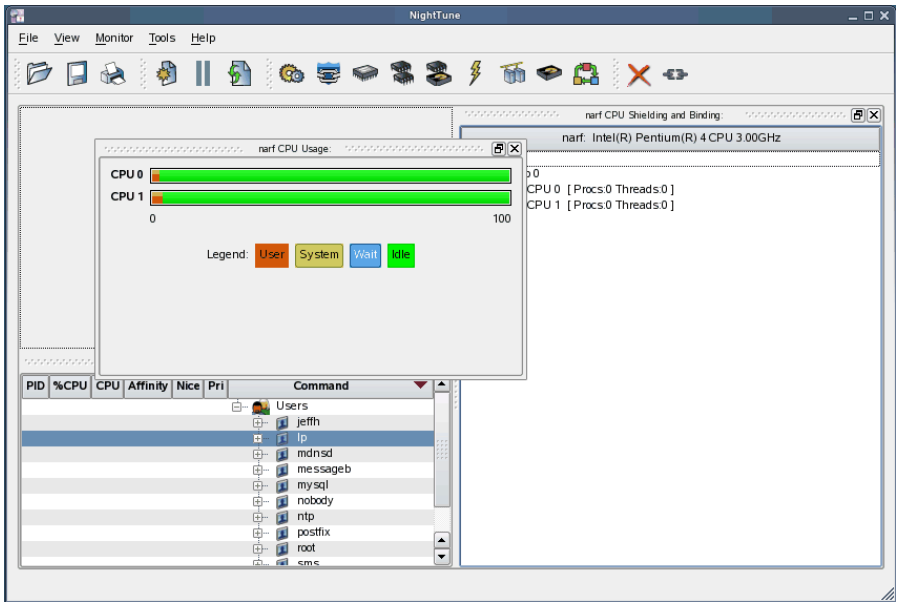
NightStar RT ツールは、移動可能かつサイズ変更可能なパネルの利用を通してニーズに合うグラフィカル・ユーザー・インターフェース柔軟に構成することが可能です。

例えば、NightTune のデフォルトの構成を考えてみて下さい。NightTune を起動した時、グラフィカル・ユーザー・インターフェースは下の図のように見えます：

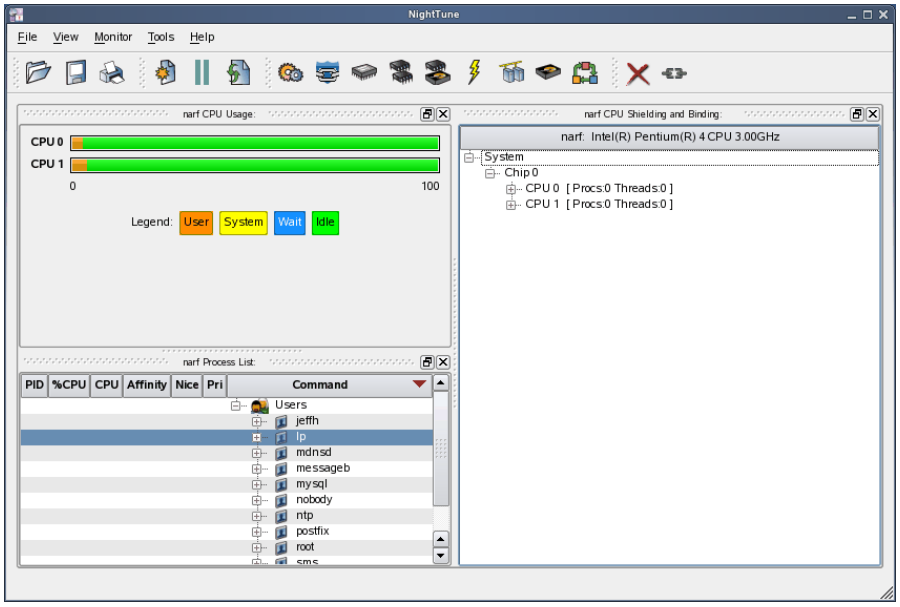


現在のページのパネルの 1 つを移動するには、移動したいパネルのタイトル・バーを左クリックして目的の場所にパネルをドラッグして下さい。アプリケーションは移動するパネルの場所に基づきそれに応じて他のパネルのサイズ変更および移動をしながらページ上の空間を生成することで応答します。

例えば、CPU Usage パネルを Process List パネルの上に移動するには、CPU Usage パネルのタイトル・バー上を左クリックしてそれを左上にドラッグして下さい。NightTune は下の図に示すように Process List パネルの上に空間を生成して応答します：

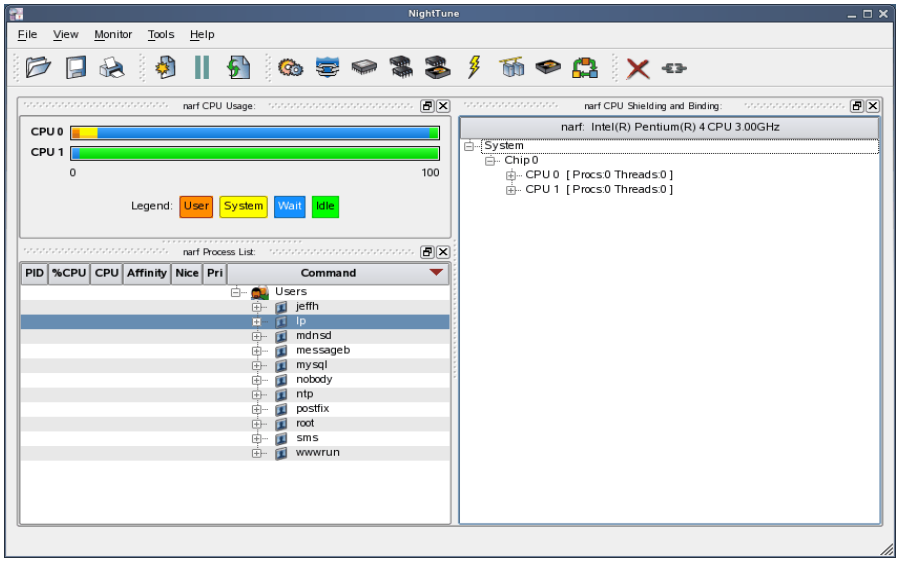


希望する場所に NightTune が空間を開いた時にマウス・ボタンを離すと NightTune はパネルをその場所に置きます。CPU Usage パネルは NightTune 画面の左上隅に常駐するようになります。



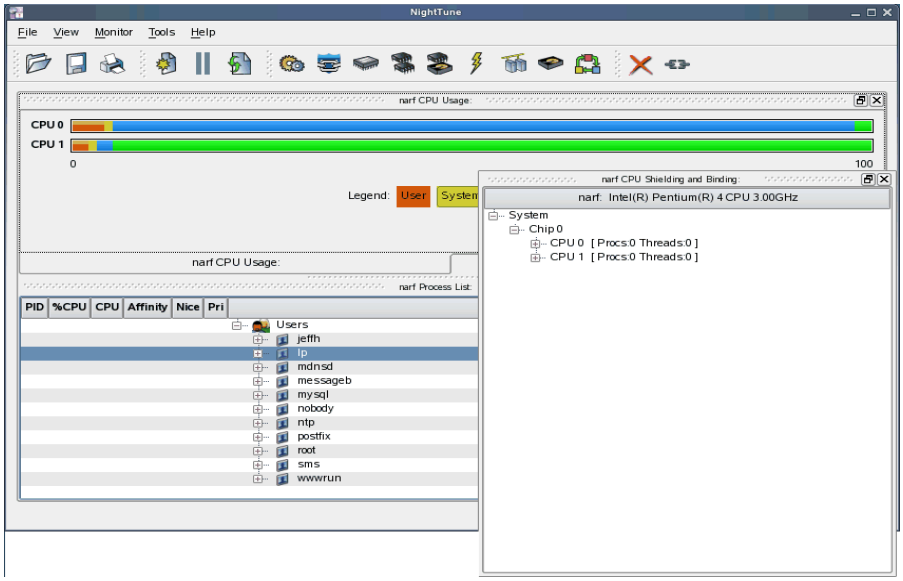
希望する場所に空き空間が現れない場合、メイン・ウィンドウのサイズを拡大し、切り離されたパネルのサイズを縮小、そして置きたい場所の近くに別の切り離されたパネルの端を移動して下さい。

パネル間のセパレーターを左クリックして希望するサイズにドラッグすることでパネルのサイズを変更することが可能です。例えば、Process List パネルの高さを上げる(結果として CPU Usage パネルの高さを下げる)には、2 つのパネルの間のセパレーターを左クリック(カーソルが両方向矢印になる)してパネルが希望するサイズになるまでセパレーターをドラッグして下さい。

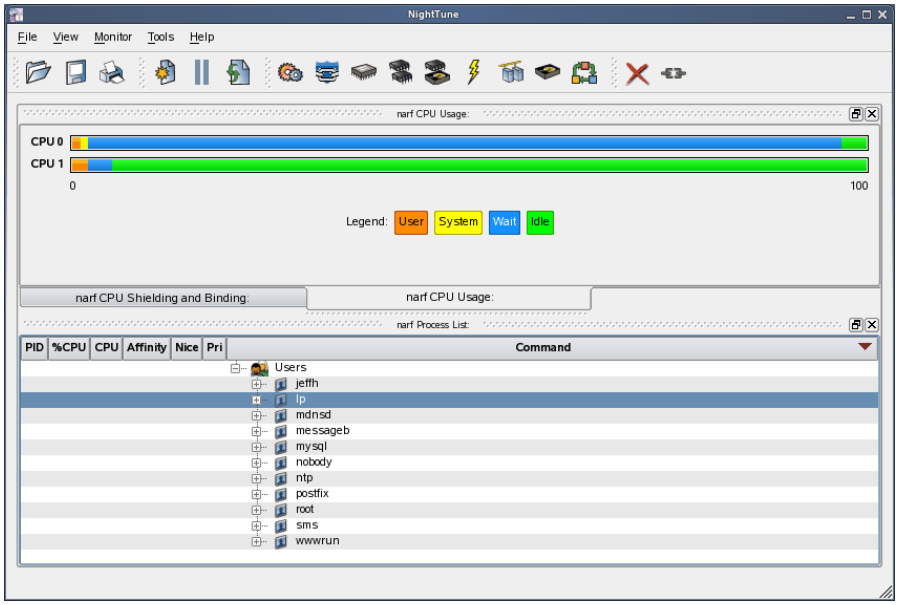


NightStar RT ツールのグラフィカル・ユーザー・インターフェースのもう一つの機能はタブ化パネルの利用となります。タブ化パネルは 2 つ以上のパネルを同じ場所に置くことで GUI の領域を最大限に生かすことが可能です。生成されたタブを利用してパネルを入れ替えることが可能です。

この例では、CPU Shielding and Building パネルと CPU Usage パネルで同じ空間を共有するように NightTune を構成します。CPU Shielding and Building パネルのタイトル・バー上を左クリックして下図に示すように CPU Usage パネルの下部に「CPU Usage」の名前が付いたタブが見えるようになるまで CPU Usage パネルの真下にドラッグして下さい。



マウス・ボタンを離すと NightTune は CPU Shielding and Building パネルを CPU Usage パネルと同じ場所に配置して、その 2 つを交互に切り替えることが可能になる 2 つのタブを真下に生成します。

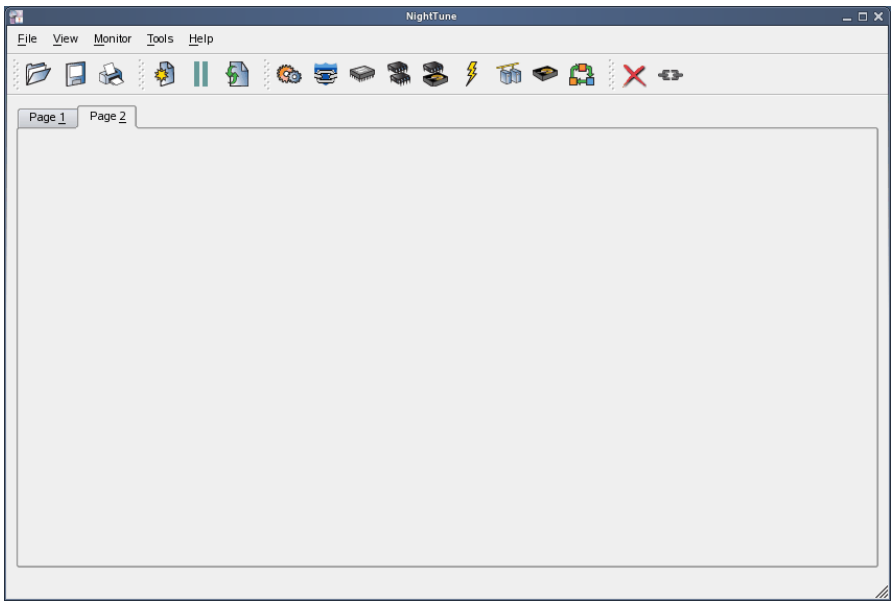


8.2. タブ化ページ

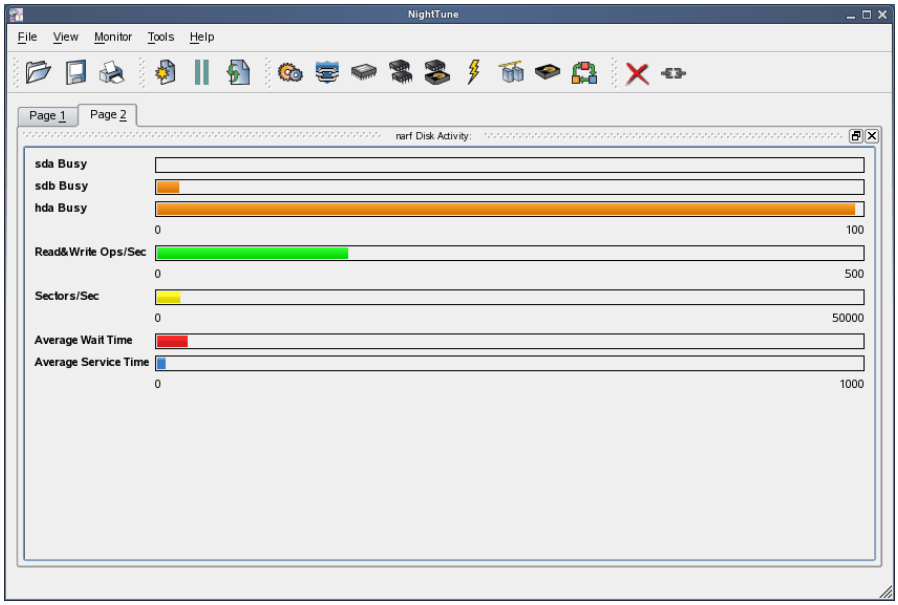
NightStar RT ツールはタブ化ページの利用を通して各アプリケーション内で複数のデータ表示とそのデータを操作するメカニズムを保持することが可能です。デフォルトで、1 つのページのみがツール起動時に表示されます。

前項の NightTune の例では、異なる一連のデータを表示する他のページを生成することが可能です。例えば、ディスク・アクティビティ、割り込みアクティビティ、メモリ・アクティビティを監視したいけど元のページを乱雑にしたくない場合があるとします。

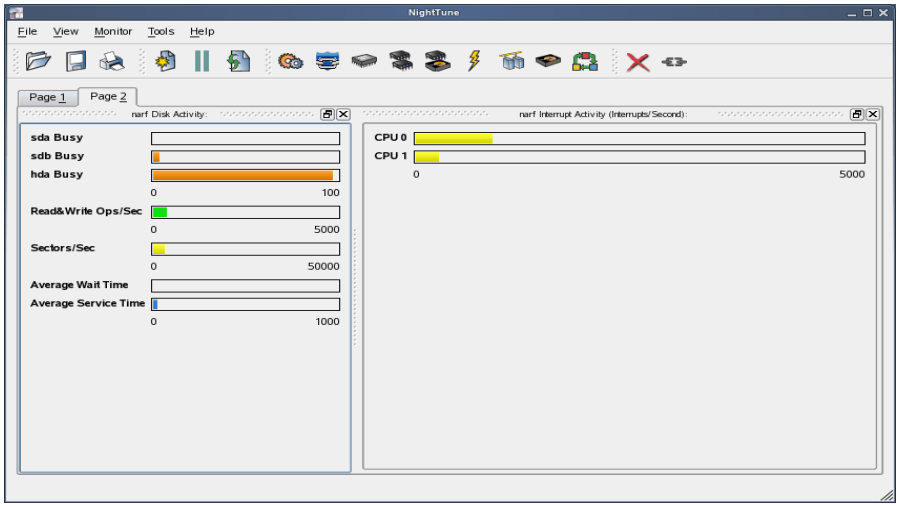
View メニューから Add Page を選択して下さい。NightTune は 2 つのタブ化ページを生成します。元のページは最初のタブの下に配置され、新しい空のページが 2 番目のタブの下に現れます。



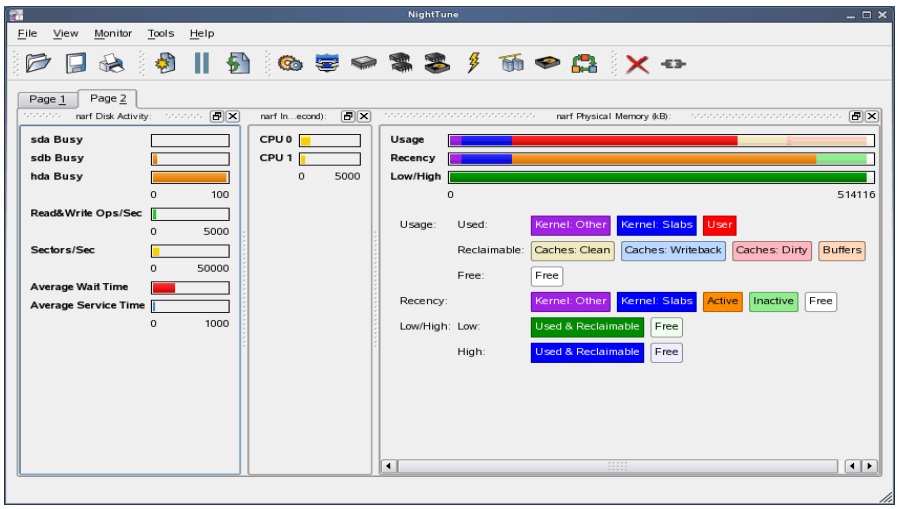
希望する NightTune パネルを追加するには、Monitor メニュー項目をクリックして下さい。選択するパネルのメニューが現れます。Disk Activity メニュー項目を選択した後にサブメニューから Bar graph pane を選択して下さい。棒グラフ形式で情報を表示する Disk Activity パネルが新しいページに追加されます。



Interrupt Activity サブメニューから Bar graph pane を選択して下さい。Interrupt Activity パネルがページに追加されます。



Memory: Physical サブメニューから Bar graph pane を選択して下さい。Memory Physical パネルがページに追加されます。



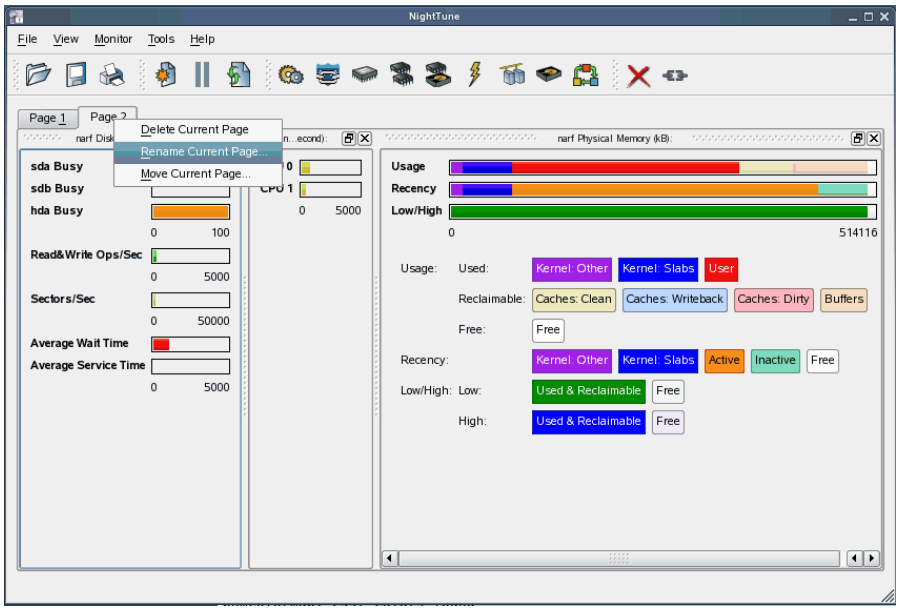
新しいページに全て棒グラフ形式情報を表示する Disk Activity, Interrupt Activity, Memory Physical パネルが収容されています。「Page 1」の名前が付いたタブをクリックすることで最初のページに復帰し、「Page 2」の名前が付いたタブをクリックすることで新しいページに戻ることが可能です。

8.3. コンテキスト・メニュー

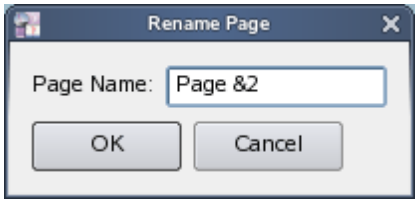
NightStar RT ツールはコンテキスト・メニューの広範囲に及ぶ利用を提供します。NightStar RT ツールの任意の場所で右クリックするとツール内のマウスの場所に関連する項目を含んだメニューをユーザーに提供します。

NightTune の例を使ってこの機能をデモします。例えば、25 ページの「タブ化ページ」で生成した新しページをより意味のある名称にしたいとします。

「Page 2」の名前が付いたタブを→クリックして下さい。メニュー項目 Delete Current Page, Rename Current Page..., Move Current Page....を持つコンテキスト・メニューが表示されます。



コンテキスト・メニューから Rename Current Page...を選択して下さい。Rename Page ダイアログが表示されます。

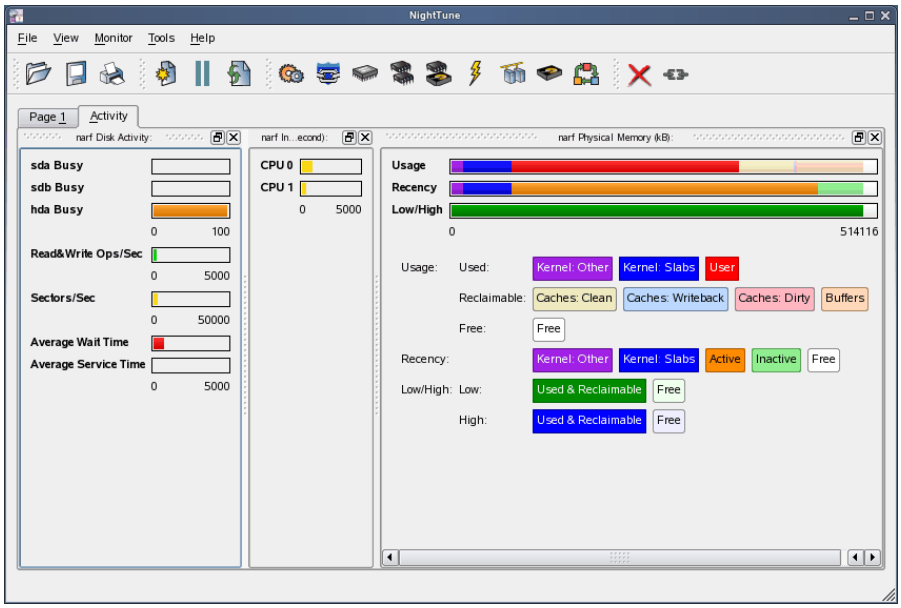


Page Name を「&Activity」に変更して下さい。

NOTE

固有文字の前のアンパサンド(&)はこのページのアクセラレーターを生成します。ユーザーは Alt キーを押しながらそのページのアクセラレーターを押すことで特定のページに切り替えることが可能です。アクセラレーターは下線がタブ上に表示されます。

元のページに切り替えるには Alt-1 を押下して下さい。Activity ページに戻るには Alt-A を押下して下さい。



9.0. NightStar RT の概要

NightStar RT ツールの各々の基本的な機能を次項で説明します。

- NightProbe
- NightSim
- NightTrace
- NightTune
- NightView
- Datamon
- Shmdefine

9.1. NightProbe

NightProbe データ・モニターの機能は次を含みます：

- プログラム・データを非干渉的にサンプリングおよび記録
- 同期および非同期でデータ・キャプチャー
- 柔軟なデータ表示機能
- サンプリング、記録、再生のための API
- 所得済みデータにタイム・スタンプ

NightProbe はプログラム、共有メモリ・セグメント、メモリ・マップ・ファイルを含むアプリケーション・リソースからデータの値を個別にモニター、修正、記録するためのツールです。NightProbe はデバッグ、解析、試作、故障注入のための開発環境、もしくはプログラムの入出力用 GUI 制御パネルを必要とする本番環境で使用することが可能です。

NightProbe はターゲット・リソースのアドレス空間を独自にマッピングする非干渉型の技術を利用しています。その後の NightProbe によるメモリの直接読み書きは、割り込みもしくはリソースへの影響なしでデータのサンプリングおよび修正を可能にします。

同期および非同期でのロギング

NightProbe は簡単な API を介してデータの同期ロギングを行うことが可能です。非同期ロギングはオンデマンド・サンプリングまたは定周期クロック・レートを用いて行うことが可能です。

NightProbe は NightTrace イベント・アナライザーによる同時解析用にトレースポイントを使用するデータ項目のロギングを提供します。アプリケーションとオペレーティング・システムの挙動が同期した図を取得するため、サンプリングしたデータはカーネル・データと付加的なユーザー・トレース・データを結合することが可能です。NightProbe はデータをディスク・ファイルに記録、または NightTrace ツールに直接データを提供することが可能です。

対話形式サンプリングおよび修正

NightProbe はオンデマンドまたはユーザー指定の更新レートでデータを定周期にサンプリングするためにフレキシブルなスプレッドシート表示を提供します。ユーザー・データの直接編集はデータ項目の新しい値をスプレッドシートに入力することで達成されます。NightProbe は個々のデータ項目に対してユーザー定義データ閾値の違反の色付け通知を提供します。

NightProbe は解析、記録、特別な表示用にサンプリングしたデータにタイムスタンプを付け、NightProbe API で書かれたユーザー・アプリケーションに渡すことが可能です。

NightProbe は静的に決定されたアドレスと形を持つ C/C++および Fortran のスカラー型および構造化データ型をサポートします。データ項目を参照、特にモニターするデータ項目の名称を入力することを許可するユーザー・プログラムのシンボル・テーブルおよびデバッグ情報を NightProbe はスキャンします。シンボル・テーブルおよびデバッグ情報を含むアプリケーションは NightProbe で利用することが可能です。アプリケーション・ソース・コードの変更は必要ありません。

9.2. NightSim

NightSim アプリケーション・スケジューラーの機能は次を含みます：

- 複数プロセスを周期的に実行
- フレーム・オーバーラン通知と制御によるメジャーおよびマイナー・サイクル
- 分散システム向け単一スケジューリング制御
- シミュレーション用アプリケーションに最適

NightSim は予測可能で周期的なプロセス実行を必要とするタイム・クリティカルなアプリケーションをスケジュールおよびモニターするためのツールです。シミュレーション用アプリケーションに最適な NightSim は開発者が動的に複数の実行、連動するプロセス、優先度、スケジューリング・ポリシー、CPU 割り当てを調整することが可能です。NightSim を利用すると、一周期の実行時間、最小、最大を表示してアプリケーションの性能をモニターすること、およびフレーム・オーバーラン発生時に処置することが可能です。

NightSim はオペレーティング・システムの Frequency-Based Scheduler (FBS)へのグラフィカル・インターフェースを提供します。これはプロセスの周期的なパターンでの実行を有効にする高分解能タスク・スケジューラーです。NightSim ではローカルもしくは分散システムで実行するプロセスのグループを簡単に構成、および生じた構成を再利用するために保存することが可能です。性能モニターは FBS で実行中のプロセスに関する CPU 使用率データを収集します。

NightSim はシミュレーション用アプリケーションの開発、デバッグ、生産段階を通して使用することが可能です。シミュレーション構成はスクリプトとして保存することが可能で、その後シミュレーションを繰り返して実行することが可能です。NightSim スクリプトは GUI 処理が禁止されているもしくは望ましくないターゲット環境で便利です。更に構成ファイルおよびスクリプトは任意のバージョン管理システムに置くことが可能です。

分散スケジューリングの同期

対象型マルチプロセッサーに加え、NigthSim は Concurrent の Real-Time Clock and Interrupt Module (RCIM)を介して接続された複数のシステムをサポートします。NightSim は分散スケジューリングの生成を単純化し、複数のターゲット・システムに渡って分散された個々のスケジューラーの同期タイミング(開始/停止/再開)を管理するための単一制御を提供します。

NightSim はリアルタイム・クロックや分散割り込みソースなどのハードウェアへのインターフェースを扱います。ユーザーはスケジューリング操作に関して土台となるオペレーティング・システムとインターフェースをとる必要はありません。

豊富な性能統計値

NightSim は、スケジュールされたプロセスごとに最大 11 個の異なる性能関連統計値、および最大 15 個の追加パラメータを監視します。最小および最大サイクル時間などの統計値を利用して、複数プロセッサー間で負荷を均衡させることによりユーザーが CPU 使用率を最適化することが可能です。NightSim の表示はカスタマイズ可能で、ユーザーが特定の統計値と監視するプロセスや重み付けされた表示の並び替え基準を選択することが可能です。

9.3. NightTrace

NightTrace イベント・アナライザーの機能は次を含みます：

- システム・アプリケーション活動状況のグラフィカルまたはテキスト表示を同期
- 単一またはマルチ・スレッド・アプリケーションでユーザー定義イベントのロギング
- システム・コール、割り込み、例外を含むカーネル・イベントのロギング
- データ解析 API
- ユーザー・コードの自動計測

NightTrace はアプリケーション、Linux オペレーティング・システムの挙動やそれらの間の相互関係を動的に表示および解析するためのツールです。NightTrace は複数の CPU またはシステムで同時に実行中の複数のプロセスからイベントを記録することが可能です。NightTrace はシステム全体の同期ビューを示すためにユーザー定義アプリケーション・イベントとカーネル・イベントを結合することも可能です。その後に NightTrace は記録された全てのログのグラフィカルな時間ベース・ビューを生成します。NightTrace ではユーザーがイベントのズーム、検索、フィルタリング、要約、解析することが可能です。トレース分析はライブもしくは実行後に行うことが可能です。

NightTrace は特にタイム・クリティカル・アプリケーションの最も厳しい要求に合うように設計されました。同期された高速ハードウェア・クロックとカーネル・フリー・プリミティブを使って NightTrace のトレースポイントは最小限のオーバーヘッドで記録されます。トレースポイントはデバイス・ドライバー、割り込みレベル・コード、任意のユーザー・アプリケーションに挿入することが可能です。トレースポイントはトレース・データを収集していない場合であっても製品品質アプリケーションの中に残しておくことが可能です。

NightTrace の Illumination ツール(nlight)は自動的に関数ごとの入り口と出口にトレースポイント(ユーザーはどの関数を装飾するのかを制御します)が付いたユーザー・コード(実行可能イメージまたは.o ファイル)を計測します。これは実行可能イメージまたは.o ファイルを完全に変更することなく行います。NightTrace は全ての引数の値や戻り値を含む関数呼び出しの解説を提供します。

グラフィックと対話形式

NightTrace は、イベントの相対タイミングを明らかに示しアプリケーションとオペレーティング・システムの活動状況の全体像を提供するため、タイムライン・グラフまたはイベント・ログと一緒に要求されたイベントと状態をグラフィカルに表示します。NightTrace は特定のイベントを探し、正確なタイミング観測のために細かい精度の等級でそれらを拡大することが可能です。NightTrace のグラフィカル表示は、カスタマイズして表示するために完全にユーザーで構成可能です。構成は保存して後で呼び出すことが可能で、複数の構成を同時に表示されることも可能です。

カーネル・トレースのサポート

割り込み、例外、コンテキスト・スイッチ、Linux システム・コール、デバイス・アクセスなどのシステム・イベント情報とユーザー・アプリケーションからのイベント情報を結合することで、NightTrace はアプリケーション実行中の任意のポイントでカーネルとユーザー・アプリケーション間の明確な相互作用の状況を提供します。

NightTrace は、頻度、発生時間、持続時間、間隔、最小/最大時間を含むイベントや状態に関する統計性能データを提供します。ユーザーは該当するプロセス、スレッド、CPU、システム、イベント内容を指定することで状態定義を生成し、イベントを分類することが可能です。条件付きトレースは C 言語式の構文を使用して示すことが可能です。オペレーティング・システムとアプリケーションの性能および動作パターンの見識を与えるために表示をカスタマイズすることが可能です。

NightTrace はアプリケーションやシステムの活動状況をモニターまたは解析する特別なアプリケーションを簡単に生成することを可能とする解析 API を使ってソース・コードを生成します。

9.4. NightTune

NightTune システムおよびアプリケーション・チューナーの機能は次を含みます：

- システムおよびアプリケーションの性能を動的に表示
- CPU 使用状況、メモリ・ページング、ネットワーク操作の監視
- プロセス、優先度、ポリシー、割り込みの対話形式制御
- プロセス、スレッド、割り込みを動的に CPU アフィニティを制御

NightTune はアプリケーションやシステムの性能をモニターおよびチューニングするためのシステム機能にグラフィカル・インターフェースを提供します。ユーザーはユーザー・アプリケーションの優先度、スケジュール・ポリシー、CPU 割り当て、CPU 使用状況をモニターすることが可能です。また、NightTune はシステムの CPU 使用状況、コンテキスト・スイッチ、割り込み、メモリ・ページング、ネットワークの動作状況もモニターします。

NightTune はプロセスを個別に、またはユーザーもしくは CPU で決定されたグループをモニターすることが可能です。また NightTune はプロセス内の個々のスレッドやタスクに関する情報も表示します。ユーザーが表示をカスタマイズすることが可能な情報を表示するのに複数のフレームおよびウィンドウが使用されます。

アプリケーションのチューニング

NightTune ではポップアップ・ダイアログやドラッグ＆ドロップ操作を使って個々のスレッド、タスク、プロセス、プロセス・グループのプロセス属性をユーザーが変更することが可能です。例えば、プロセス・アイコンを CPU アイコンにドラッグするとプロセスはそのプロセッサーにバインドします。その後ユーザーはグラフと文字の両方でチューニングの試みの結果を即座に確認します。

システムのチューニング

NightTune ではポップアップ・ダイアログやドラッグ＆ドロップ操作を使って割り込みの CPU 割り当てをユーザーが変更することが可能です。NightTune はオプションで、NightTune のセッション中の全てのアプリケーションおよび実行されたシステム・チューニングのテキスト・ログを提供します。

9.5. NightView

NightView ソース・レベル・デバッガーの機能は次を含みます：

- 単一インターフェースを介してマルチ・システム、マルチ・プロセッサー、マルチ・プロセス、マルチ・スレッドをデバッグ
- ブレークポイント、モニターポイント、ウォッチポイントを含む活線パッチ
- アプリケーション速度に適用
- 動的メモリ「ヒープ」のデバッグ
- 実行中の変数の変更および表示

NightView ではユーザーが複数のタイム・クリティカル・プロセスを同時にデバッグすることが可能です。NightView の使用で、プログラマーはプログラムを停止または割り込みせずにプログラムを変更、実行、修正またはデータを表示することが可能です。ヒットおよび無視カウントなどのイベントポイント条件はアプリケーション内に直接パッチが適用され、完全なアプリケーション速度で実行することが可能です。NightView はアプリケーション・タイミングに悪影響を与えることなくきめ細やかな制御を提供します。

NightView のモニタポイントはプロセスを止めることなくユーザーが選択した場所で式を表示することが可能ですので、提供するデータはアプリケーションのアルゴリズムに同期して表示します。ウォッチポイントはユーザー指定の変数またはメモリ位置が選択的に読まれたもしくは変更された場合にアプリケーションを停止するハードウェア・アドレス・トラップ機能を利用します

言語依存のデバッグ

NightView は C/C++、Fortran の任意の組み合わせで書かれた複数のアプリケーションのデバッグをサポートします。各プログラム内の全ての変数と式は適切な言語で参照されます。また NightView は NightTrace イベント・アナライザーと統合されています。NightView は NightTrace による同時もしくは実行後の解析用にユーザー指定の位置にトレースポイントを挿入することが可能です。

Gnu デバッガーよりも更に強力

NightView は gnu デバッガー(**gdb**)では利用できない多くの機能を用意しています。NightView が優位な点は、ユーザーが単一セッションから複数のプロセスおよびスクリプトから開始されたプロセスをデバッグする機能を含んでいます。NightView では、パッチが当てられたコードを完全な速度で実行します。プロセスが実行中、活線パッチは変数の変更またはイベントポイントの追加が可能です。モニタポイントは式やスタック変数の表示が可能で、デバッガーを迂回してシグナルをプロセスに直接送信することが可能です。

動的メモリのデバッグ

NightView はコードの再コンパイルなしでデバッグ処理中にメモリの問題を見つけて取り除く手助けをする対話形式メモリ・デバッガーを含んでいます。NightView はヒープ・メモリのリークを監視し、アプリケーションが使用するメモリの量をモニターし、どのようにメモリを割り当てて解放したのかを追跡します。このメモリ・デバッガーを有効にすると、NightView はヒープの割当てと解放をリアルタイムでユーザーに追跡させるので、実行後の解析よりも更に効率よくデバッグすることが可能です。プログラマーは実行を停止し、問題を確認し、パッチを試験して、そしてデバッグを続行することが可能です。NightView は二重解放、未解決ポインター、ヒープ領域の超過、その他一般的なユーザー・アプリケーションのバグを検出することが可能です。

9.6. Datamon

Datamon は単独で実行中のプロセスをリアルタイムでユーザーが変数を監視、記録、修正することが可能なユーザー・アプリケーション・インターフェースです。これは適格な変数を調べるためにプログラム・ファイルをスキャンし、型名、アトミック型、ビット・サイズ、ビット・オフセット、形状、コンポーネント・メンバー、アドレスを含む属性に関する詳細な情報を取得する機能を含んでいます。Datamon は変数のアクセスおよび変更について非干渉の手法を利用しています。

9.7. Shmdefine

Shmdefine は独立するプログラム間でのデータの共有を支援します。コモン・ブロックの共有は Fortran プログラム間で最も便利ではありますが、それは Fortran、C 言語、Ada のプログラムで IPC 共有メモリ・サービスを効果的に利用するのに役立ちます。

10.0. はじめに

NightStar RT *Tutorial* を NightStar RT 製品の入門書として***強く推奨***します。本チュートリアルは、NightStar RT ツールの全てを豊富な機能を実演する様々なシナリオを具体的にして 1 つにまとめた実例に統合しています。

チュートリアルは *NightStar RT Installation DVD* の `documentation` ディレクトリおよびインストール後の**`/usr/share/doc/NightStar/pdf`** に PDF フォーマットで存在します。

チュートリアルのオンライン・バージョンは、デスクトップにインストールされた NightStar RT Documentation アイコンをダブルクリックして `Bookshelf` から NightStar RT `Tutorial` を選択することでアクセスすることが可能です。

加えて、チュートリアルは任意の NightStar RT ツールの Help メニューから起動、または次のコマンドをコマンド・ラインから実行することで開始することが可能です：

```
nhelp
```

10.1. ケーパビリティ

NightStar RT の殆どの操作で特別な特権を必要としません。しかしながら、root ユーザーとして実行することなく NightStar RT の機能を最大限に利用したい場合、追加の構成手順が必要となります。

Linux は、ある特権を持つ操作を実行するための権限を特権のない別のユーザーに付与する手段を提供します。Pluggable Authentication Module(**`pam_capability(8)`**を参照)は様々な作業で必要になる一連のケーパビリティ(*ロール*と呼ぶ)を管理するために使用されます。

以下の表は、NightStar RT の使用で推奨されるケーパビリティを持つ非 root ユーザーに付与される利点を示します：

ケーパビリティとその効果

ケーパビリティ	利点
`CAP_IPC_LOCK`	ユーザー・トレース・イベントのバッファに関連するメモリにクリティカル・ページをロックすることを NightTrace に許可します。
`CAP_SYS_RAWIO`	PCI デバイスや**`/dev/mem`** などのメモリ・マップされたシステム・ファイルへアクセスすることを NightProbe に許可します
`CAP_SYS_NICE`	スケジューリング・ポリシー、スケジューリング優先度、プロセスの CPU アフィニティを設定することを NightStar RT ツールに許可します。 割り込みの CPU アフィニティを設定、およびプロセス、割り込み、ハイパースレッドの干渉から CPU をシールドすることを NightTune に許可します。

RedHawk がインストールされたシステムでは `CAP_SYS_NICE, CAP_SYS_RAW_IO, CAP_IPC_LOCK` ケーパビリティを提供する *`nstaruser`* ロールが既に構成されているはずです。

/etc/security/capability.conf を編集して(まだ定義されていない場合は)nstaruser ロールを「ROLES」セクションに定義して下さい：

```
role nstaruser CAP_SYS_NICE CAP_IPC_LOCK CAP_SYS_RAWIO
```

更に、ターゲット・システムの各 NightStar RT ユーザーごとにファイルの最後に次の行を追加して下さい：

```
user username nstaruser
```

username はユーザーのログイン名称です。

ユーザーが nstaruser ロールで定義されていないケーパビリティが必要な場合、nstaruser を含む新しいロールと必要になる追加のケーパビリティを書き加えて、上述の nstaruser の代わりに新しいロール名称を指定して下さい。

/etc/security/capability.conf にログイン名称を登録することに加えて、**/etc/pam.d** ディレクトリ下の特定のファイルについても有効にするためのケーパビリティを許可する構成にする必要があります。

WARNING

本項の残りで **/etc/pam.d** 内のファイルを編集するように求められます。対応する共有ライブラリがシステムに実際に存在することを確信していない場合はこれらの変更を行わないで下さい。さもないと再びログインすることが出来ない可能性があります。

これらのファイルの編集で間違いがあった場合に簡単に回復できるようにこれらの操作を行う場合はターミナル・セッションを root として実行したままにすることは良い考えです。例えば root でシステムに **ssh** でログインし、ファイルを編集して **sshd** デーモンを再開した場合、現在の root セッションは失われ、再度ログインすることが出来なくなる可能性がありますので注意して下さい。

ケーパビリティを有効にするには、**/etc/pam.d** 内の選択したファイルの最後に次の行を(存在しない場合は)追加して下さい：

```
session required pam_capability.so
```

変更するファイルのリストは、システムにアクセスするために使用される方法のリストに依存しています。以下の表は、システムへのアクセスに使用される最も一般的なサービスについてユーザーにケーパビリティを許可する推奨の構成を示しています。

推奨する/etc/pam.d 構成

/etc/pam.d ファイル	影響するサービス	コメント
common-session	殆ど全て	より新しいシステム(例：Ubuntu 16.04)で、本ファイルは殆どのサービスに含まれています。その場合、(他のテーブルは無視して)単にこのファイルにエントリを追加して下さい。
remote	telnet rlogin rsh (コマンド無しで使用時)	お手持ちのシステムに依存しますので、**remote** ファイルが存在しない可能性があります。**remote** ファイルは生成せず、存在する場合にのみ編集して下さい。
password-auth	殆ど全てのログイン・メカニズム	本ファイルは最近の OS ディストリビューションに存在します。これが存在する場合、本ファイルに前述の内容を追加して下さい。
login	local login (例：コンソール) telnet * rlogin * rsh * (コマンド無しで使用時)	* Linux の一部のバージョンでは、**remote** ファイルの存在は **login** ファイルの範囲をローカル・ログインに制限します。このような状況で、**login** と共にここに示されている他のサービスは **remote** 構成ファイルのみに影響します。
rsh	rsh (コマンド使用時)	例： **rsh system_name a.out**
sshd	ssh	**/etc/ssh/sshd_config** も同様に編集して次の行が存在することを確認する必要があります： **UsePrivilegeSeparation no**
gdm	gnome セッション	
lightdm	Mate セッション	
kde	kde セッション	
systemd-user	一部の GUI セッション	

/etc/pam.d/sshd または**/etc/ssh/sshd_config** を編集する場合、変更の効果を得るため、基本的な OS のバージョンにもよりますが次のコマンドのいずれかを使って **sshd** サービスを再起動する必要があります：

- **/sbin/systemctl restart sshd**

上記の変更の効果を得るには、ユーザーはログオフしてターゲット・システムに再度ログインする必要があります。

許可したケーパビリティを確認するには、次のコマンドを実行して下さい：

```
getpcaps $$
```

このコマンドからの出力は、現在割り当てられているロールがリストアップされます。

このコマンドが利用できない場合、次のコマンドを実行すると CapPrm と CapEff で始まる行に非ゼロの数値を確認できるはずです。表示されたこれらの 16 進数は個々のケーパビリティに関するビット・マスクです。

```
cat /proc/self/status | egrep -e "^Cap"
```

```
CapInh: 0000000000000000
CapPrm: 0000000000824000
CapEff: 0000000000824000
```

10.1.1. NightView にプロセスへのアタッチを許可

デフォルトで最近の一部の Linux ディストリビューションは、例えデバッグしたいプロセスと同じ UID や GID でデバッガーを起動したとしても実行中のプロセスにデバッガーをアタッチさせないように制限しています。

制限を制御する変数の値を変更する **sysctl** コマンドを使用してこの制限を取り除くことが可能です。制限を取り除くには、シェルから次のコマンドを入力してください：

```
sysctl -w kernel.yama.ptrace_scope=0
```

このコマンドを実行して、ログアウトし再度ログインしたら制限が解除されます。しかしながら、設定は次のリブートまでのみの効果となります。システムを起動する度に適用されるようにするには上記コマンドを**/etc/rc.local**(時には**/etc/rc.d/rc.local**)の中に置いても構いません。

11.0. NightStar RT のライセンシング

NightStar RT は NightStar RT ツールへのアクセスを制御するのに NightStar License Manager (NSLM) を使用します。

ライセンスのインストールは Concurrent Real-Time が提供するライセンス・キーが必要です。NightStar RT ツールはライセンス・サーバー(41 ページの「ライセンス・サーバー」を参照)からライセンスを要求(41 ページの「ライセンスの要求」を参照)します。

購入された製品のオプション次第ですが、2 つのライセンス・モード(固定とフローティング)が利用可能です。固定ライセンスはローカル・システムから NightStar RT ユーザーに供給されることのみが可能です。フローティング・ライセンスはネットワーク上の任意のシステムで任意の NightStar RT ユーザーに供給することが可能です。

ツールはシステムごと、同時ユーザーごとに許可されます。同じシステムから同じユーザーによる一部または全ての NightStar RT ツールの使用は、自動的に 1 つのライセンスを共有します。この意図は、*n* 人の開発者が *n* 個のライセンスを必要とする間に同時に全てのツールを完全に利用することを可能にするためです。ツールはローカル・システムで起動しますがリモート・システムと対話するリモート・モードでツールが動作する場合、ライセンスはホスト・システムからのみ必要となります。

ローカル・システムにインストールされた全てのライセンス、現在の使用状況、デモ・ライセンスに関する有効期限をリストアップするライセンス・リポートを取得することが可能です(41 ページの「ライセンス・リポート」を参照)。

デフォルトのオペレーティング・システムの構成は、フローティング・ライセンスに干渉する場合のある厳格なファイアウォールを含む可能性があります。構成などの対応に関する情報については 42 ページの「フローティング・ライセンス向けファイアウォール構成」を参照して下さい。

11.1. ライセンス・キー

ライセンス・モード(固定またはフローティング)に応じてローカルまたはリモート・クライアントのどちらかに供給するために特定のシステムにライセンスが許可されます。

ライセンスのインストールは Concurrent Real-Time が提供するライセンス・キーが必要です。ライセンス・キーを取得するには、システム識別コードを提供する必要があります。システム識別コードは **`nslm_admin`** ユーティリティにより生成されます：

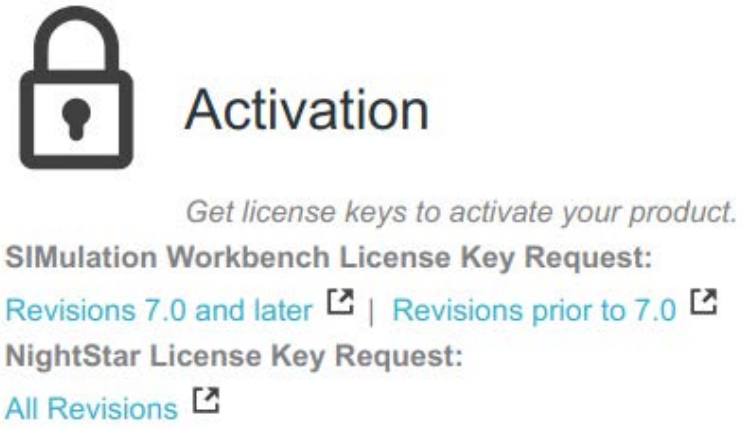
```
nslm_admin --code
```

IMPROTANT

システム識別コードはシステム構成に依存します(詳細は 40 ページの「ライセンス用ネットワーク・デバイスの選択」を参照)。システムに Linux または NightStar RT を再インストールもしくはネットワーク・デバイスの交換は新しいライセンス・キーの取得が必要となる可能性があります。

ライセンス・キーを取得するには、次の URL 利用して下さい：

https://www.concurrent-rt.com/customer-support



図： ライセンス・アクティベーションのリンク

上図に示すように **NightStar License Key Request** の下にある「All Revisions」をクリックして下さい。

システム識別コードを含む要求される情報を提供して下さい(但し、40 ページの「ライセンス用ネットワーク・デバイスの選択」を最初に読んで下さい)。ライセンス・キーがすぐにメールで送信されます。

次のコマンドを使ってライセンス・キーをインストールして下さい：

```
nslm_admin --install=xxxx-xxxx-xxxx-xxxx-xxxx
```

xxxx-xxxx-xxxx-xxxx-xxxx はライセンス承認メールに含まれているキーとなります。

必要な情報がすぐに利用できないもしくは特別な事情がある場合、コンカレント日本に連絡して下さい(詳細については 51 ページの「ソフトウェアの直接サポート」を参照して下さい)。

11.1.1. ライセンス用ネットワーク・デバイスの選択

デフォルトで、**nslm** はシステムに問合せて利用可能な全てのネットワーク・デバイスを調査します。**nslm** は実際のネットワーク・デバイスを仮想デバイスを超えて選択しますが、これは後者が常に利用できないことがあるためです。

次のコマンドを使って **nslm** が使用可能なネットワーク・デバイスのリストを参照することが可能です：

```
nslm_admin --devices
```

デフォルトで、**nslm** はリスト内の最初のデバイスを選定します。

ライセンス・キーに関連付ける特定のデバイスを指定したい場合、コードを取得する際に **--device** オプションを追加して下さい：

```
nslm_admin --device=eth2 --code
```

11.2. ライセンスの要求

デフォルトで、NightStar RT ツールはローカル・システムからライセンスを要求します。ライセンスが利用できない場合、システムのホスト名の IP アドレスに関連付けられたローカル・サブネットにライセンス要求をブロードキャストします。

/etc/nslm.config 構成ファイルを使ってシステム全体のライセンス要求を制御することが可能です。

デフォルトで**/etc/nslm.config** ファイルは次のような行を含んでいます：

```
:server @default
```

引数**@default** はコロン区切りのシステム名、システム IP アドレス、ブロードキャスト IP アドレスに変更することが可能です。ライセンスは、ライセンスが許可されるもしくはリスト内の全てのエントリーが使い尽くされるまでリストにある各構成要素から要求されます。

例えば、次の設定はローカル・システムのみを指定することでライセンスのブロードキャスト要求を止めます：

```
:server localhost
```

次の設定は**server1**からライセンスを要求し、ライセンスが供給されない場合は次に**server2**、その後はブロードキャスト要求をします：

```
:server server1:server2:10.134.30.0
```

同様に **NSLM_SERVER** 環境変数を使ってツールの個々の起動でライセンス要求を制御することが可能です。設定した場合、上述のとおりコロン区切りのシステム名、システム IP アドレス、ブロードキャスト IP アドレスを含む必要があります。**NSLM_SERVER** 環境変数の使用は **/etc/nslm.config** で定義された設定よりも優先します。

11.3. ライセンス・サーバー

NSLM ライセンス・サーバーは NightStar RT インストール時に自動的にインストールされ実行するように構成されます。

nslm サービスは自動的にランレベル 2, 3, 4, 5 で有効になります。次のコマンドを実行するとこれらの設定を確認することが可能です：

- **/usr/bin/systemctl status nslm**

滅多にはありませんが、次のコマンドでライセンス・サーバーを再起動する必要がある可能性があります：

- **/usr/bin/systemctl restart nslm**

詳細については**nslm(1)**を参照して下さい。

11.4. ライセンス・リポート

ライセンス・リポートは **nslm_admin** ユーティリティを使って取得することが可能です。

```
nslm_admin --list
```

ローカル・システムにインストールされた全てのライセンス、現在の使用状況、(デモ・ライセンスの)有効期限をリストアップします。**--verbose** オプションの使用でライセンスが現在許可されている個々のクライアントもリストアップします。

--broadcast オプションの追加でシステムのホスト名に関連付けられたローカル・サブネット上でブロードキャスト要求に応答する全てのサーバーに関するこの情報をリストアップします。

オプションや詳細については **nslm_admin(1)**を参照して下さい。

11.5. フローティング・ライセンス向けファイアウォール構成

デフォルトの Red Hat 構成はフローティング・ライセンスに干渉する厳格なファイアウォールを含みます。

このようなシステムがライセンスの供給に使用される場合、サーバー要求を通過させるために少なくとも 1 つのポートがそのファイアウォールでオープンされる必要があります。詳細については 42 ページの「フローティング・ライセンス向けファイアウォール構成」を参照して下さい。

同様にそのようなシステムが NightStar RT ツールのホストである場合、ライセンス・サーバーからライセンスを受信できるように少なくとも 1 つのポートがそのファイアウォールでオープンされる必要があります。これが行われない場合、ツールはフローティング・ライセンスを要求するツールは受信できず正しく機能しません。詳細については 43 ページの「ファイアウォールの環境で NightStar RT ツールを実行」を参照して下さい。

11.5.1. ファイアウォールの環境でライセンスを供給

以下は NSLM ライセンス・サーバーが実行中のシステムでファイアウォールが構成されている場合にそれがフローティング・ライセンスを供給可能にするいくつかのアプローチです：

- システムのファイアウォールを完全に無効
- 特定のシステムからの NSLM ライセンス要求を許可
- 特定のサブネット上の任意のシステムからの NSLM ライセンス要求を許可
- 任意のシステムからの NSLM ライセンス要求を許可

NOTE

ファイアウォール構成を変更するには root である必要があります。

これらの指示は RHEL5 の **iptables** のバージョンが対象です。新しいバージョンでは異なる構成手法、異なるファイアウォール・スキーマに置き換えられている可能性があります。いずれにしても、下に表示された情報は概念を理解するのに役立つ可能性があります。

ファイアウォールを完全に無効にするには次を実行して下さい：

```
service iptables stop
```

続いて**/etc/sysconfig/iptables** ファイルを削除して下さい：

```
rm -f /etc/sysconfig/iptables
```

このオプションは見た目よりも危険ではありません。大抵は、ネットワーク全体がファイアウォールで保護されているので、ネットワーク上の個々のシステムで更に保護する必要はありません。不安な場合、ネットワーク管理者に確認して下さい。

残りのケースにおいて、簡単な変更を**/etc/sysconfig/iptables** ファイルに行う必要があります。デフォルトで、ファイルは次のような行を含んでいるはずです：

```
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
```

特定のシステムからの NSLM ライセンス要求を許可するには、REJECT 行の前に次の行を挿入して下さい：

```
-A RH-Firewall-1-INPUT -p udp -m udp -s system --dport 25517 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp -s system --dport 25517 -j ACCEPT
```

これらの行は複数のシステムで繰り返すことが可能です。

特定のサブネット上の任意のシステムからの NSLM ライセンス要求を許可するには、REJECT 行の前に次の行を挿入して下さい：

```
-A RH-Firewall-1-INPUT -p udp -m udp -s subnet/mask --dport 25517 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp -s subnet/mask --dport 25517 -j ACCEPT
```

subnet は `192.168.1.0` のような書式で、*mask* は `255.255.255.0` のような伝統的なネットワーク・マスクまたはマスクの一部を左からのビット数で示す `24` のような単一番号です。例えば、`192.168.1.0/255.255.255.0` と `192.168.1.0/24` は同等です。

これらの行は複数のサブネットで繰り返すことが可能です。

任意のシステムからの NSLM ライセンス要求を許可するには、REJECT 行の前に次の行を挿入して下さい：

```
-A RH-Firewall-1-INPUT -p udp -m udp --dport 25517 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 25517 -j ACCEPT
```

/etc/sysconfig/iptables の変更後に次を実行して下さい：

```
service iptables restart
```

11.5.2. ファイアウォールの環境で NightStar RT ツールを実行

以下は NightStar RT ツールが実行中のシステムでファイアウォールが構成されている場合、NightStar RT ツールがライセンス・サーバーからフローティング・ライセンスを受信可能にするいくつかのアプローチです：

- 要求するシステムのファイアウォールを完全に無効
- 特定のライセンス・サーバーからの NSLM ライセンスを許可
- 特定のサブネット上の任意のシステムからの NSLM ライセンスを許可
- 任意のシステムからの NSLM ライセンスを許可

NOTE

ファイアウォール構成を変更するには root である必要があります。

ファイアウォールを完全に無効にするには次を実行して下さい：

```
service iptables stop
```

続いて**/etc/sysconfig/iptables** ファイルを削除して下さい：

```
rm -f /etc/sysconfig/iptables
```

このオプションは見た目よりも危険ではありません。大抵は、ネットワーク全体がファイアウォールで保護されているので、ネットワーク上の個々のシステムで更に保護する必要はありません。不安な場合、ネットワーク管理者に確認して下さい。

残りのケースにおいて、簡単な変更を**/etc/sysconfig/iptables** ファイルに行う必要があります。デフォルトで、ファイルは次のような行を含んでいるはずです：

```
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
```

ライセンス・サーバーが実行中の特定のシステムからの NSLM ライセンスを許可するには、REJECT 行の前に次の行を挿入して下さい：

```
-A RH-Firewall-1-INPUT -p udp -m udp -s server --sport 25517 -j ACCEPT
```

これらの行は複数のサーバーで繰り返すことが可能です。

特定のサブネット上のライセンス・サーバーが実行中の任意のシステムからの NSLM ライセンスを許可するには、REJECT 行の前に次の行を挿入して下さい：

```
-A RH-Firewall-1-INPUT -p udp -m udp -s subnet/mask --sport 25517 -j ACCEPT
```

subnet は `192.168.1.0` のような書式で、*mask* は `255.255.255.0` のような伝統的なネットワーク・マスクまたはマスクの一部を左からのビット数で示す `24` のような単一番号です。例えば、`192.168.1.0/255.255.255.0` と `192.168.1.0/24` は同等です。

これらの行は複数のサブネットで繰り返すことが可能です。

ライセンス・サーバーが実行中の任意のシステムからの NSLM ライセンスを許可するには、REJECT 行の前に次の行を挿入して下さい：

```
-A RH-Firewall-1-INPUT -p udp -m udp --sport 25517 -j ACCEPT
```

/etc/sysconfig/iptables を変更後に次を実行して下さい：

```
service iptables restart
```

以下は NSLM ライセンス・サーバーが実行中のシステムでファイアウォールが構成されている場合にそれがフローティング・ライセンスを供給可能にするいくつかのアプローチです：。

- システムのファイアウォールを完全に無効
- 特定のシステムからの NSLM ライセンス要求を許可
- 特定のサブネット上の任意のシステムからの NSLM ライセンス要求を許可
- 任意のシステムからの NSLM ライセンス要求を許可

NOTE

ファイアウォール構成を変更するには root である必要があります。

ファイアウォールを完全に無効にするには次を実行して下さい：

```
service iptables stop
```

11.6. ライセンス・サポート

ライセンス問題の追加支援については、コンカレント日本に連絡して下さい。詳細については 51 ページの「ソフトウェアの直接サポート」を参照して下さい。

12.0. アーキテクチャの相互運用性

NightStar RT ツールはセルフ・ホスト環境だけでなく、タイム・クリティカルなターゲット・システムからホスト処理を分離してリモートからでも使用できるように設計されました。

12.1. i386 と x86_64 アーキテクチャの問題

NightProbe

制限なし

NightSim

制限なし

NightTune

制限なし

NightTrace

NightStar RT 5.0 でのアーキテクチャ間の機能

- 32-bit アプリケーションは NightTrace Logging API を使って 64-bit システムで実行することが可能です。64-bit の NightTrace は **`ntraceud`** や **`ntrace`** を介してそのようなプログラムからのデータをキャプチャーし解析することが可能です。32-bit アプリケーションは本リリース(またはそれ以降)のバージョンの NightTrace Logging API にリンクさせる必要があります。
- 64-bit システムで実行中の NightTrace は 32-bit システムで生成されたデータ・ファイル(ユーザーとカーネルの両方)を解析することが可能です。しかしながら、ユーザー・トレース・データを生成する 32-bit アプリケーションは、NightStar RT 5.0 ベース(またはそれ以降)のバージョンの NightTrace Logging API にリンクさせる必要があります。
- NightTrace Logging API を使用する 64-bit アプリケーションは、ストリームまたはファイル・モードのどちらでも 32-bit アプリケーションから生成されたデータを解析することが可能です。

ユーザーの責任

- 64-bit システムで 32-bit データを解析する場合、64-bit システムで評価するのと同じように NightTrace は式のタイプを評価します。従って、NightTrace の式で明確に `arg_long()`を指定すると 4byte のみが記録されたとしてもトレース・データからは 8byte のデータが抽出されることになります。データ型の懸念は `long` と全ての `pointer` 型です。

NOTE

NightTrace は自動的に正しいデータ型で **Timelines** と **Event** パネルに引数を出力します。

- NightTrace 内で 32-bit で生成されたブロック引数を解凍する際、32-bit コンパイラーが構造体を配列したものとして解凍する必要があります。`long`, `long double`, 全ての `pointer` 型のサイズが異なることに加えて、64-bit コンパイラーは構造体に異なるパディングを挿入しますので注意して使用して下さい。これには 32-bit Application Illumination セッションで生成された情報の使用が含まれています。(`long` の参照は 4byte だけを期待していたとしても 8byte を抽出します。)

  NOTE

  実際には、`long double` 項目が 32-bit アプリケーションから生成された場合に 64-bit NightTrace セッションで `arg_long_dbl()`を使用しても問題ありません。64-bit システムで `long double` の後ろに余分な 4byte のデータがありますが、これらの余分なバイトはそのような値を操作する命令によって(現在は)完全に無視されます。

NightView

制限

以前のバージョンでは、NightView は 64-bit マシーンで 32-bit アプリケーションをデバッグするには--arch=i386 オプションが必要でした。この制限は撤廃されました。--arch=i386 オプションは本リリースでは効果がありません、無視されます。例え 32-bit x86 プログラムを exec する(**`exec(2)`**を参照)プログラムが混在していたとしても 64-bit x86 システムで 32-bit x86 プログラムを自由にデバッグすることが可能です(64-bit シェルから起動される 32-bit プログラムを含みます)。

12.2. Intel と ARM64 の相互運用性

NightTrace

x86_64 または aarch64 システムでキャプチャーしたバイナリの NightTrace データ・ファイルは x86_64 または aarch64 システムのどちらからでも解析することが可能です。「*NightStar RT 5.0 でのアーキテクチャ間の機能*」項で説明する制限が同様に適用されます。

NightSim

制限なし

NightProbe

制限なし

NightTune

制限なし

NightView

X86 と ARM64 はアーキテクチャに互換性がないため、aarch64 システムで x86 の実行可能プログラムを実行もしくはその逆も出来ません。

しかしながら、NightView は x86_64 から aarch64 ターゲット・システムへの「クロス・デバッグ」は可能です。これは x86_64 ホストに **`ccur-nview-aarch64-support`** パッケージがインストールされ、aarch64 ターゲットに **`ccur-nview-target`** パッケージがインストールされている必要があります。

完全なクロス・デバッグのサポート(例えば、パッチ、条件付きイベントポイント等)については、GNU x86/aarch64 クロス開発パッケージも同様にホスト・システムにインストールする必要があります。パッケージの最小セットは次のとおり：

- gcc-aarch64-linux-gnu
- binutils-aarch64-linux-gnu

これらのパッケージは CentOS 系および Ubuntu 系のシステムで大抵は利用可能です。

現在、NightView は次の場所にクロス・コンパイラーがあることを期待します：

`/usr/bin/aarch64-linux-gnu-gcc`

お使いのディストリビューションが他の場所に置いている場合、実際のコンパイラーへのシンボリック・リンクを作成する必要があります。その後 NightView 内からのクロス・コンパイラーへのパスの設定が許可されるよう NightView は変更されます。

GNU クロス開発環境を使って完全なユーザー・アプリケーションをビルドする場合、追加パッケージが必要になる可能性があります。一般的なクロス開発については本資料の範囲外となります。

13.0. 既知の問題

本項では NightStar の既知の問題を列挙します。

13.1. 問題：位置独立実行形式

NightStar は位置独立実行形式(Position Independent Executables: PIE)をまだ完全にはサポートしていません。最新の Ubuntu システムでは PIE はデフォルトです。次の機能は PIE 実行ファイルを現在利用できません：

- NightProbe
- Datamon
- Application Illumination (nlight)

NightStar は次のメジャー・リリースで PIE を完全にサポートします。それまでは、以下の gcc と ld のオプションの組み合わせを使って非 PIE 実行ファイルをビルドすることが可能です。

```
gcc -fno-PIC -fno-PIE -no-pie ...
```

NOTE

gcc 4.9.3 を使用する Ubuntu 20.04 は、これを防ぐためにコンパイラーに与えたオプションに関係なく現在は PIE 実行ファイルのみをビルドします。

13.2. 問題：Ubuntu 20.04 - NightView ロード・コマンド

これは PIE 問題の延長です。Ubuntu 20.04 において全てのオブジェクトは PIE を使用します。NightView は Ubuntu 20.04 上で.o ファイルのロードをまだサポートしていません。

13.3. 問題：ターゲット・システムにアタッチ出来ない

いくつかの NightStar ツールは動作するためにサーバー・プロセスと通信する必要があります。

デフォルトで、次のコマンドで返される値を使ってローカル・システムのサーバー・プロセスを探そうとします：

```
hostname
```

ホスト名へのマッピングがない場合、これらのツールは機能しなくなります。

解決策 1:

`/etc/hosts` が有効な IP アドレスへのホスト名のマッピングを含んでいる、またはマッピングが DNS もしくは他の手段で利用可能になっていることを確認して下さい。

解決策 2:

SELinux の強制モードを permissive に変更して下さい。**`/etc/selinux/config`** ファイルを編集してこの変更を行い再起動して下さい。NightStar 開発チームはこの問題の他の解決策に取り組んでいます。

13.4. 問題：NightView がメモリ・セグメントにマッピングできない

一部のシステムでは、SELinux がインストールされると enforcing モードが設定されます。ディストリビューションの種類やバージョンにもよりますが、これは NightView がアプリケーションのメモリ・セグメントへマッピングするのを邪魔する可能性があります。

解決策:

SELinux の強制モードを permissive に変更して下さい。**`/etc/selinux/config`** ファイルを編集してこの変更を行い再起動して下さい。NightStar 開発チームはこの問題の他の解決策を検討しています。

14.0. ソフトウェアの直接サポート

ソフトウェアのサポートがセントラル・ソースから利用可能です。システムに関する支援または情報が必要な場合、コンカレント日本の技術サポートサービス部(**03-3864-5717**)まで連絡して下さい。営業時間は祝祭日を除く平日の 9 時から 17 時までとなります。

また、コンカレント日本の Web サイト **https://www.concurrent-rt.co.jp/** を介して、または電子メールを **tech@concurrent-rt.co.jp** 宛てに送信することでいつでも支援要請を依頼することも可能です。